1

A Model of Global Positioning System (GPS) Master Control Station (MCS) Operations

THESIS

David Nicholas Koster
Captain, USAF

AFIT/GSO/ENS/92-9

AFIT/GSO/ENS/92-9

A Model of Global Positioning System (GPS) Master Control Station (MCS) Operations

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Space Operations

David Nicholas Koster, B.S.E.E.
Captain, USAF

December 15, 1992

Approved for public release; distribution unlimited

Thesis Approval

STUDENT: Capt David N. Koster *CLASS:* GSO-92D

THESIS TITLE: A Model of Global Positioning System (GPS)
Master Control Station (MCS) Operations

DEFENSE DATE: November 25, 1992

COMMITTEE:	*NAME/DEPARTMENT*	*SIGNATURE*
Co-Advisor:	Thomas F. Schuppe, Col, USAF Assistant Professor of Operations Research	Thomas F Schuppe
Co-Advisor:	Kenneth W. Bauer, Jr., Lt Col, USAF Associate Professor of Operations Research	Kenn W Bauer Jr.

Acknowledgements

Many people helped me perform this research, create this document, and succeed at obtaining this lofty (for me) degree. Some of those are generally unaware of or do not understand their great contributions:

- My parents, Donald and Earlena Koster, who supplied the better part of my motivation. Perhaps the better part of me, period. The desire to make them proud has never left me.
- My family, Debra and Leslie, who gave the love and peace of mind that allowed me to study (and sleep) in untroubled domestic tranquility. Of course, I was gone so often and for so long, they couid just as well pretended I was TDY.
- My fellow students, who provided academic and social support, even when I didn't deserve it. I gave them what I had, which sometimes was useful but most often a distraction.

Those who are very aware of their contribution deserve my thanks, also. Like Col Thomas Schuppe and Lt Col Ken Bauer, my advisors, who supplied encouragement and motivation. I wanted them to be proud, too. Thanks to Lt Col TS Kelso, who is Space Operations at AFIT; I wish we'd seen more of you. Thanks to the folks at the MCS, see you soon.

Finally, to the junior Space Ops jockeys and any other who read this: be good, have fun, and although my body's in Colorado, my spirit's still in Glen Helen; come visit me sometime.

[NOTE: The document was prepared primarily on Sun SPARCstations using LaTeX, *BibTeX*, Bill Ogden's *BibCard*, *Xfig*, and *GNUplot*.]

David Nicholas Koster

AFIT/GSO/ENS/92-9

Abstract

The United States Air Force's Navstar Global Positioning System (GPS) provides high-accuracy space-based navigation and time distribution to suitably-equipped military and civilian users. The system consists of earth-orbiting satellites and a world-wide network of ground stations. A single operational control center, the GPS Master Control Station (MCS) monitors, maintains, and commands the GPS satellite constellation. The on-going deployment of the complete satellite constellation and recent changes in the operational crew structure may invalidate previously used planning and management paradigms. There is currently no analytical method for predicting the impact of these and other environmental changes on system parameters and performance. Extensive testing cannot be performed at the MCS itself due to the criticality of the GPS mission and lack of operational redundancy. This research provides and validates a discret event simulation model of the MCS operations center task flow, focusing on the creation and testing of a sliding-widow MCS activity scheduler. The simulation was validated using MCS historical data. Experiments were conducted by varying the number of ground stations and satellite constellation size available to the simulation. The results, while not quantitatively trustworthy, were used to draw general conclusions about the GPS operational environment.

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

Table Page

Table Page

A Model of Global Positioning System (GPS) Master Control Station (MCS) Operations

I. Introduction.

1.1 Problem.

The United States Air Force's Navstar Global Positioning System (GPS) is rapidly approaching fully operational status. The mission of the GPS is to provide high-accuracy space-based navigation and time distribution to suitably equipped military and civilian users (5:1). The system consists of a growing number of earth-orbiting satellites, a world-wide network of ground stations and communications segments, and a single operational control center, the GPS Master Control Station (MCS). The MCS, its ground stations and support facilities, and the communications network needed to interconnect these facilities, constitute the GPS Operational Control System (OCS).

The MCS monitors, maintains, and commands the GPS satellite constellation. It operates continuously with a crew of eight active-duty Air Force personnel. Their primary responsibilities are maintaining GPS time and navigation accuracy and periodically assessing the functional health of each GPS space vehicle (SV) (5:11). Continuous monitoring of satellite navigation data transmissions, forwarded to the MCS from five remote Monitor Stations (MS), allow the system positioning and timing accuracy to be evaluated. Four GPS Ground Antennas (GAs) provide the primary means of directly contacting SVs to upload improved navigation data, collect SV telemetry, and command configuration changes.

The on-going deployment of the complete satellite constellation and other changes in the operational crew structure may invalidate previously used planning and management paradigms.

There is currently no analytical method for predicting the impact of these and other environmental changes on system parameters and performance. Extensive testing cannot be performed at the MCS itself due to the criticality of the GPS mission and lack of operational redundancy. One solution to this problem is to create a simulation model of the MCS operational activity and perform the needed testing and experimentation using this tool. This will allow both planned and unplanned events affecting the MCS to be assessed without impact to current operations. This thesis provides and validates such a model and uses it to perform experiments on the probable operational impact of changes to the GPS operational environment.

1.2 Proposed Solution.

A model will be created, validated, and used to perform experiments in order to predict the perfoımance of the Global Positioning System Operational Control System under a number of operational environment conditions. The model could allow GPS management to predict system resource loading under circumstances not yet experienced by the system. It could also allow "what-if" analysis to be performed to determine the response of mission effectiveness criteria to predicted system configuration changes.

1.3 Methodology.

In detail, this thesis will:

- Describe and evaluate current literature on the Global Positioning System organizational structure and operations, system modeling, simulation, and scheduling,
- Determine measurable criteria for evaluating the performance of the GPS MCS. MCS output products such as system computer and crew logs, operational performance measures, and interviews with GPS operations staff and crews will be used to develop these criteria,
- Evaluate competing methodologies for producing a model of the operations task flow at the MCS, based on the data gathered and the established performance criteria,
- Develop a model of current GPS MCS operations, including (as a minimum) the effects of:
  - Satellite contact requirements,

  - Satellite constellation size,
  - Ground station availability,
  - MCS crew availability and operational constraints, and
  - Unscheduled OCS resource outages.

- Validate the MCS model against current operational data not used in the development of the model,
- Create performance baselines for both the current system configuration and the planned 24 satellite operational constellation,
- Perform the experiments using the model and evaluate the results against the current performance baselines.

1.4 Implementation

It was determined that a computer simulation was the best model type for this task. Factors leading to this decision were the iterative nature of the scheduling technique to be used and the need to emulate the stochastic MCS activity durations. SIMSCRIPT II.5 (a registered trademark of CACI Products Company) was the simulation language selected to implement the model. The organization of the model program was tailored to correspond to the functional organization of the MCS, to aid in development and performance verification. To validate the completed simulation, a program environment was created to duplicate the conditions that existed at the MCS on 0000Z on 10 April 1992. The simulation was then executed and its scheduling performance was compared the actual MCS Master Contact Schedule for that day. The result was that 35.09% (20 of 57) of the satellite activities scheduled by the simulation were within ten minutes of the historical scheduled time, and 31.58% (18 of 57) exceeded one hour. On this basis, the model was judged to be insufficiently accurate to allow direct quantitative comparison to MCS performance. However, experimentation was performed to allow a qualitative analysis of the model's performance under various conditions.

1.5 Results.

The results of experimentation indicate the model may be useful for predicting the general behavior of the Global Positioning System Master Control Station operational performance. The relationships between MCS resource scheduling capability and mission requirements were characterized, as were the effects of satellite constellation growth and ground station outage. While the results are not useful in predicting actual resource utilization rates, they do indicate what can be expected in terms of relative utilization and rate of increased resource use.

II. Background

2.1 Chapter Overview.

The creation of a simulation requires a comprehensive understanding of two systems. The first is the system being modeled. These objects include the people operating the system (if any), the devices used (again, if any), and the resources the system consumes or produces. The system being modeled here is the Global Positioning System, which requires all the above objects. This chapter will describe this system in detail, using information from GPS program development documentation, actual system operating instructions, and first-hand GPS operational experience.

The second system requiring in-depth familiarity is the modeling environment itself. Creation of an effective model requires knowledge of general simulation and operations research theory; the relative advantages of alternative modeling systems; and in-depth knowledge of the environment selected for use. These topics as they pertain to this research are covered in this chapter.

2.2 The Navstar Global Positioning System.

The Navstar Global Positioning System (GPS) is the United States Air Force's space-based, highly-accurate radionavigation system. Since it was first made available for three-dimensional position determination in December 1978, this system has created a revolution in disciplines such as navigation, ordinance delivery, search and rescue, surveying, and precise timing (10:1-7). The GPS allows any number of properly-equipped users to simultaneously determine their time to within 100 nanoseconds of Universal Time; their position within 16 meters spherical error probable (SEP) (5:8), and their velocity to within 0.2 meters per second (10:1-13).

Very simply, GPS consists of a number of earth-orbiting satellites that continuously broadcast the current time and their orbital parameters. Users equipped with GPS receiver sets on land, sea, in the air, and in low-earth orbit, receive this information from all the satellites in their view. Sign and data processing components in the receiver set estimate the current position of each satellite and measure the propagation delay of radio signals from the satellite to the user. Using these time delays, the set solves simultaneous equations (one for each satellite used) in position and time to produce the *users* time, position, and velocity.

The accuracy of the above calculations depends on knowing the satellite time and position precisely. However, the satellite on-board time standard and the satellite orbits are subject to both random and cumulative error. GPS satellites are currently not able to make autonomous and real-time corrections to the time and position data they broadcast, so a network of monitoring receivers continuously collect satellite range data. These data are transmitted to a master control station, where accurate estimates of satellite time standard drift and orbital trajectory are calculated. These estimates are mathematically propagat_d into future, formed into the proper data format, and transmitted back to the satellites via transmitting antennas. The satellite then retransmits this data to the users.

As a system, the GPS consists of three major subsystems (Figure 2.1). The Space segment consists of the satellite constellation. The Control Segment is the Master Control Station, the monitoring and transmitting stations, and the communications links between them. Finally, the User Segment is the community of military and civilian organizations and personnel equipped with GPS receiver sets. The functional inter-relationship of these segments is diagrammed in Figure 2.2. Each segment is described in detail in the following sections.

2.2.1 Space Segment. The Space Segment is a growing number of earth-orbiting satellite vehicles (SVs), which constitute the GPS "constellation". An understanding of function of the satellites and the implications of the geometry of their orbits is required for correct model design and analysis.

2.2.1.1 GPS Satellite Orbits. The orbital height for operational GPS SVs is nominally 10,930 nautical miles, or about 20,200 kilometers. This orbital altitude gives the satellites an 11 hour

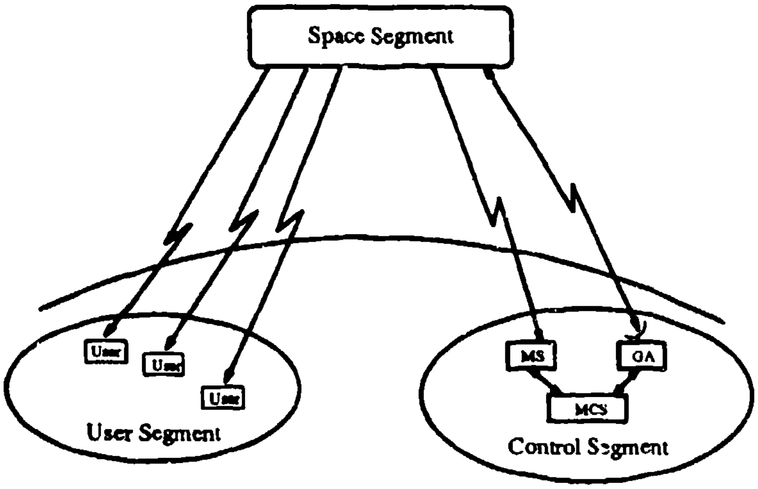

Figure 2.1. Navstar GPS Segments

58 minute (semi-synchronous) orbital period. When combined with the motion of the Earth, this orbit results in each satellite appearing in the same location of the sky once per day for a fixed observer. However, the difference between the satellite orbital period and the length of the solar day causes each satellite to appear approximately 4 minutes earlier each day. The intended orbits are circular; in practice, most have an eccentricity of less than 0.01. There are six orbital planes, labeled "A" through "F", into which the satellites are launched. There will be four satellite in each plane, spaced 90 degrees apart. As noted below, the orbital planes of the research and development SVs have a greater inclination than that of the operational satellites.

The orbital parameters of the GPS constellation were chosen to optimize the performance of the navigation mission. There are a number of tradeoffs involved. High altitude orbits provide long periods of visibility for a given location on the Earth, but these distances require greater transmitted signal power for reliable communication. At lower altitudes, however, the SV/user range changes rapidly as the satellite passes overhead, degrading system accuracy.

2.2.1.2 GPS Satellites. From February 1978 to September 1985, ten GPS research and development satellites (collectively called "Block I") were launched. These satellites, designed and

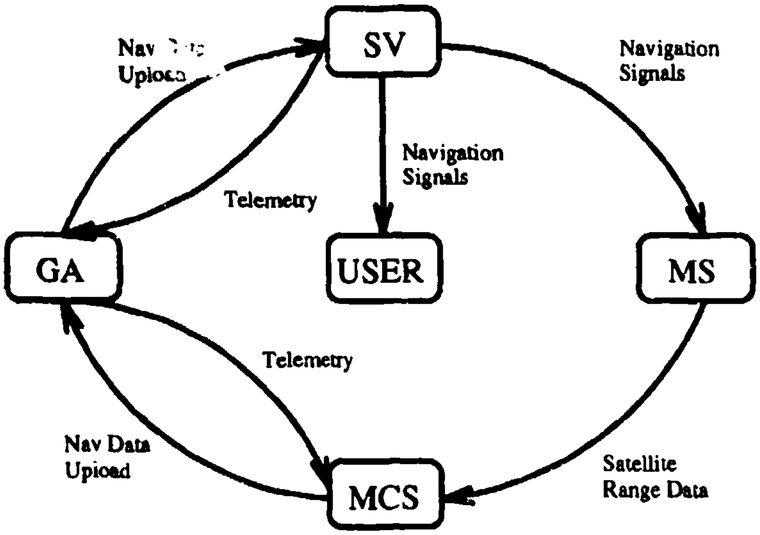

Figure 2.2. Global Positioning System Functional Diagram

built by Rockwell International, were designated BI (for Block I) -1 through -11. (The numbering discrepancy arises from the launch failure of BI-5, whose Atlas booster exploded on the launch pad in December 1981 (10:1-8).) These spacecraft, intended to have a five-year lifespan, allowed the GPS concept to be tested and validated. The Block I satellites have shown remarkable longevity, however, and four of these SVs were still providing service as of July 1992 (3). The eldest of these, BI-8, was launched in July 1983 (10:1-8). Block I GPS satellites have an orbital inclination of 64 degrees, as opposed to 55 degrees for the operational constellation. This was to improve satellite visibility over the test sites (primarily the Yuma Proving Ground in Arizona) during the system test and evaluation phase (10:1-6).

The launching of operational satellites began in February 1989 and continues to the present. These satellites, labeled Block II, in addition to being larger and more expensive, have additional capabilities over the Block I SVs. While the Block I on-board satellite navigation processors had only enough memory to retain 26 hours of navigational data, the Block II SVs can store 14 days of usable data. The Block II vehicle electronics suite was hardened against radiation, the effects of which plagued the Block I vehicles. The design life of Block II GPS SVs is 7.5 years (5:91).

A subsequent upgrade to the Block II vehicles, designated Block IIA, further increased the usable on-board navigation data storage to 180 days. The Block II SVs are numbers BII-1 to BII-9; BII-10 was the first Block IIA satellite. The Block II satellites can be programmed to prevent users without official authorization (in the form of cryptographic keys) from using the system to its fullest capability.

The current target is to have a constellation of 21 operational satellites with three on-orbit spares (5:7). The spares will be active, bringing the minimal number of usable satellites for an operational system up to 24 (10:1-3). This number does not include the remaining operational Block I satellites. The upper limit to the number of operational satellites is 30, due to the design of the Master Control Station (MCS) (5:11). A 24 satellite constellation will result in the 98% three-dimensional positioning availability requirement specified in the System Operational Requirements Document (SORD) (10:6-2).

As a secondary payload, the later Block I and all subsequent satellites carry nuclear burst detection sensors, part of the Nuclear Detonation (NUDET) Detection System (NDS). These sensors detect the detonation of nuclear devices anywhere in the world and relay the exact time and range of the burst to suitably equipped receivers. This data, hopefully from multiple satellites, is then used to calculate the precise time and location of the detonation.

2.2.2 User Segment. Military and civilian users who require precise time or position information and are equipped with suitable GPS receivers make up the User Segment. As described above, each satellite continuously broadcasts information that allows GPS receivers to calculate the satellite's position with respect to a common reference system. With information from three satellites, the GPS receiver can compute ("triangulate", in a sense) the users' position and velocity in three dimensions. However, because the receiver set's clock must be synchronized with the GPS common time, the information from a fourth satellite is needed to derive the users' local clock/GPS time offset. If the user knows (or does not care to know) any part of the time and position equation, the remaining unknowns can be calculated with data from fewer than four satellites. Two examples of this are ships at sea (their height above sea level is known) and users with highly accurate time standards synchronized to GPS time. These users would require information from only three satellites.

A important byproduct of this process is that the user now has an extremely accurate time reference, as part of the of data received from the satellite describes the offset between GPS time and Universal Coordinated Time, or UTC. These data allow suitably equipped users to calculate the local time to within 100 nanoseconds (100 billionths of a second) of UTC.

There are two levels of GPS accuracy: the Precise Positioning Service (PPS) and the Standard Positioning Service (SPS). PPS users are those with both a suitably-equipped user set and the necessary cryptological keys to decode the PPS transmissions from the GPS satellites. Users with PPS-incompatible receivers, and authorized users without cryptokeys, are restricted to the SPS. PPS provides positional accuracy with errors no greater than 16 meters Spherical Error Probable (three dimensional error not to exceed 16 meters), while SPS provides 76 meters SEP. Both types of users have access to precise time and time interval information (5:7).

Receiver sets vary in size and sophistication dependent on the requirements of the users. For instance, the Small, Lightweight GPS Receiver (SLGR) hand-held set weighs less than five pounds and will fit in the pocket of a standard-issue Battle Dress Uniform (10:1-10). This user set can receive only the SPS signals, which provides better than 100-meter horizontal error for footsoldiers and slow-moving vehicles. Larger and more complex receivers, with the ability to electronically track up to five SVs simultaneously, are designed to mount in high-performance aircraft and have the computing ability to update a fast-moving users' position quickly. As described above, authorized users receive cryptographic keys to decode the highest accuracy signals from the GPS satellites.

2.2.3 Control Segment. Lastly, the Control Segment consists of a central control facility called the Master Control Station (MCS), a number of transmitter and receiver stations located around the world to gather system data and command and control the satellite constellation, and world-wide communications network that conveys data to and commands from the MCS. The Control Segment has the responsibility for maintaining system availability and accuracy within established standards. Using the dedicated communications network and ground stations, the MCS continuously collects data broadcast by GPS satellites, processes the data into accurate satellite time and location predictions, then transmits the predictions back to the satellites. This data in turn is rebroadcast to the GPS users, who use the data to solve for their own time and location. At the same time the calculated data is transmitted to the satellite, the operational crew at the MCS collects and reviews satellite telemetry data for satellite anomaly detection and performance trending. The Control Segment is the focus of this research and is described in detail in the following sections.

2.3 GPS Operational Control Segment.

The 2nd Satellite Operations Squadron (2 SOPS), located at the Consolidated Space Operations Center (CSOC), Falcon Air Force Base (AFB), Colorado, has the mission of operating the GPS. Falcon AFB is itself located approximately 15 miles east of the city of Colorado Springs, Colorado. The 2 SOPS, a subordinate unit of the 50th Space Wing, is also responsible for maintaining the GPS ground stations, maintaining navigation and timing quality control/quality assurance, and providing user interface for special users. The GPS Control Segment, also called the Operational Control System (OCS), consists of five Monitor Stations (MS), three standard Ground Antennas (GA), a Prelaunch Compatibility Station (PCS), the MCS, and the communications network needed to link the MCS with the remote stations. The OCS also may utilize the ground antenna (part of the Air Force Satellite Control Network (AFSCN)) located at Falcon AFB. This GA has the AFSCN network designation "PIKE". Each of these type of facilities is described below, with Table 2.1 providing the exact locations. Figure 2.3 is a block diagram of the entire OCS.

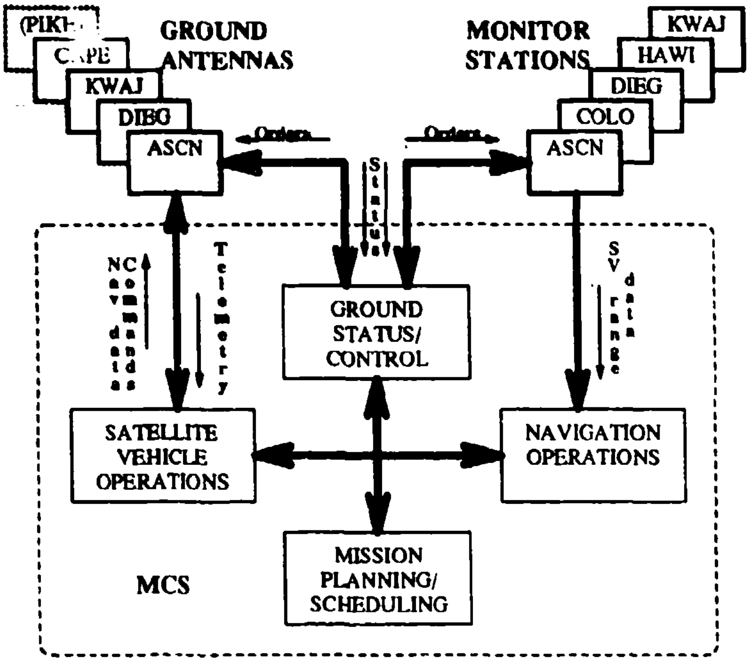

Figure 2.3. GPS Operational Control Segment (OCS) Block Diagram (from (5:119)).

2.3.1 Monitor Stations. Monitor Stations serve as the MCS's listening posts for the navigation signals transmitted by the satellites overhead. There are five GPS Monitor Stations, four scattered about the globe near the equator and one co-located with the MCS at Falcon AFB. (refer to Table 2.1 for specific MS location data). Monitor Stations are sophisticated, semi-autonomous receivers, able to track the signals of up to nine satellites simultaneously under command of the MCS. The MCS provides the MS with information on the satellites in view, allowing data preprocessing at the MS to reduce the amount of data returned to the MCS and to speed the satellite signal acquisition process.

Monitor Stations process the signals received from the SVs and continuously calculate the range between the MS antenna and each SV being tracked. The MS antenna locations have been surveyed to within one meter to accomplish this task with the precision required (5:87). The MS transmits the results of these measurements to the MCS. The MCS then uses these data to accurately estimate the positions of the SVs and the drift of their on-board time standards. Another critical role of the MS is to provide immediate feedback to the MCS in the case of satellite failure or anomalous behavior. The MCS crew watches the availability and quality of satellite data from the Monitor Stations closely for any sign of navigation signal degradation or satellite subsystem failures.

Table 2.1. GPS Monitor Station and Ground Antenna Locations (5:106)

Designation	Location	Type	Latitude (deg)	Longitude (deg)
ASCN	Ascension Island, South Atlantic Ocean	MS/GA	7.9S	14.4W
COSPM	Falcon AFB, Colorado	MS	38.8N	104.5W
PIKE	Falcon AFB, Colorado	GA*	38.8N	104.5W
DIEG	Diego Garcia Island, Seychelles, Indian Ocean	MS/GA	7.2S	72.4E
KWAJ	Kwajalein Atoll, West Pacific Ocean	MS/GA	8.7N	167.7E
HAWAIM	Oahu, Hawaii	MS	21.6N	158.2W
CAPE	Cape Canaveral AFB, Florida**	GA	28.5N	80.6W

* AFSCN Remote Tracking Station
** Prelaunch Compatibility Station

2.3.2 Ground Antennas. The GPS Ground Antennas are the primary means for command and control of GPS satellites. The three dedicated GPS GAs are co-located with Monitor Stations at Kwajalein, Ascension, and Diego Garcia, as described in Table 2.1. The Prelaunch Compatibility Station (PCS) at Cape Canaveral is an identical facility used for ensuring the interoperability of new satellites with the OCS just prior to launch. When not reserved for its testing role, it serves as a fully functional GA (5:95).

In addition to these dedicated resources, the OCS can use the AFSCN Remote Tracking Station (RTS) located at Falcon AFB for communicating with the GPS constellation. This is achieved through the use of a unique electronic interface device that allows the RTS to emulate a GPS Ground Antenna. As this facility is not dedicated to GPS and has substantial AFSCN commitments, its use must be prearranged with AFSCN schedulers. The AFSCN operators can perform command and control functions on GPS satellites, but that network is not configured to calculate and upload navigational data.

Each GPS GA is an active S-band transmitter and receiver, equipped with a 10-meter diameter steerable parabolic dish antenna. On command from the MCS, the antenna is pointed using the current MCS estimate of the satellite position. Once two-way contact is established with the SV, the GA transmits real-time or prestored commands and simultaneously receives SV telemetry (5:95). Figure 2.4 shows schematically what the relative GA coverage areas are, given the GA longitude and an approximate 140 degree coverage arc (5:99).

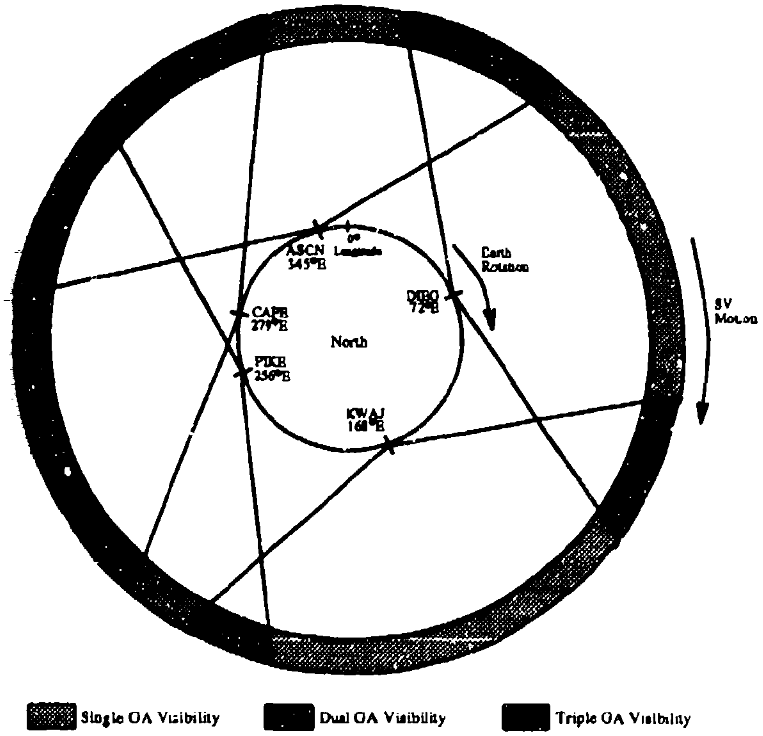

Figure 2.4. GPS Ground Antenna Coverage (from (5:99)).

2.3.3 Communications. Control Segment communications link the MS/GAs to the MCS. There are generally two 9600 bit-per-second (bps) links to each of the GAs, and a single 9600 bps link to the MSs. The communications channels are carried by both military and commercial communications networks (5:100).

2.3.4 Master Control Station. The MCS acts as the hub for all GPS activities. Its mission is to command and control the GPS satellite constellation and the other GPS resources. The MCS is designed to monitor and control 30 GPS satellites simultaneously in any mix of Block I and Block II types (5:11). The MCS is one of four satellite operations centers in the CSOC. Physically, it consists of the operations center ("ops center") itself, the MCS computer facility, maintenance and support facilities, the Colorado Springs Monitor Station, and an NDS mission support area operated by the Air Force Technical Applications Center (AFTAC), the primary users of the NDS data.

In the ops center, there are 11 computer console positions. Any of the six distinct crew functions (listed in Table 2.2) can be performed at any of the consoles. Detailed descriptions of the responsibilities of each crewmember are provided below.

Table 2.2. GPS MCS Crew Position Summary

Position	Symbol	#	Rank
Flight Commander	FCMDR	1	Lt/Maj
Crew Chief	CCHIEF	1	TSgt-SMSgt
Satellite Analysis Officer	SAO	1	Lt/Capt
Satellite Vehicle Officer	SVO	1	Lt/Capt
Satellite Support Operator	SSO	3	Amn/TSgt
Ground Systems Operator	GSO	1	Amn/TSgt

The MCS computer center, using two IBM 3083 mainframe computers and associated storage and communications hardware, is the heart of the operations center. The computers, together with the GPS Mission software, provide all data processing support for the operations mission. The major services this system provides includes:

- data collection from the Monitor Stations,
- calculation of satellite trajectory and satellite and MS time standard drift estimates,
- creation and formating of SV navigation uploads,
- maintenance of system parameter databases and logs,
- operations consoles display format and control,
- support for transmission of GPS data to external agencies, and
- support for internal functions such as planning, scheduling and report generation.

The tasks assigned to each MCS crew position provide a comprehensive overview of how the MCS performs its command and control mission. These tasks are described below. The general crew task flow and interaction is shown graphically in Figure 2.5.

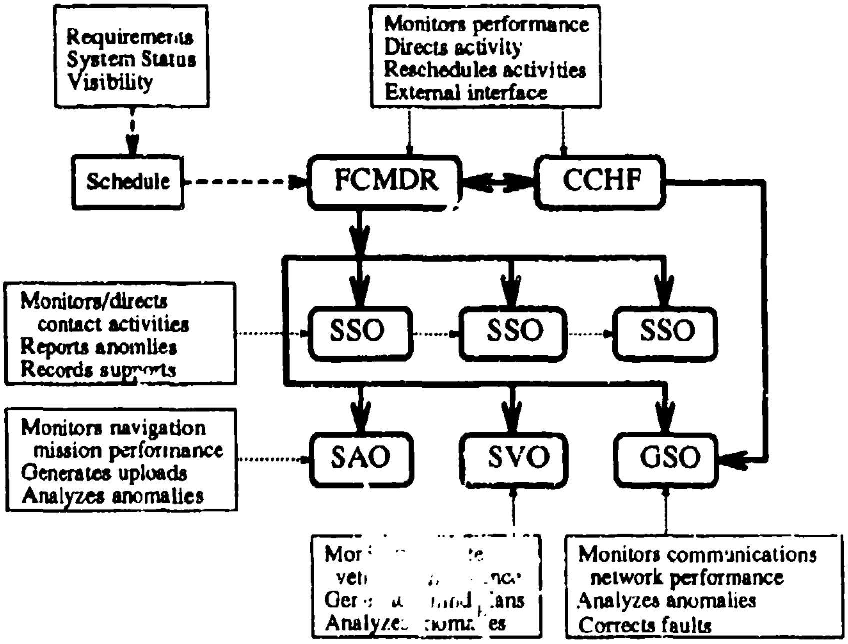

Figure 2.5. MCS Crew Organization and Tasks (5:37)

2.3.4.1 Flight Commander (FCMDR). The FCMDR monitors and directs the crew in the performance of mission operations. The FCMDR approves the schedule for his shift, makes schedule changes as required, and gives the final authorization for the preparation and execution of every satellite contact. The FCMDR is also the focal point for communications between the crew and external agencies.

2.3.4.2 Crew Chief (CCHIEF). The CCHIEF, in addition to acting as the deputy Flight Commander, is responsible for coordinating the activities of the MCS with maintenance functions and other support activities. The CCHIEF also performs the up-channel reporting tasks assigned to the operational crew.

2.3.4.3 Satellite Support Operator (SSO). The SSO personnel perform the satellite contacts. This entails coordinating and monitoring GA operations, configuring the system for the contact, executing the commands directed by the schedule, and recording the details of the contact. The SSO also reviews satellite telemetry during the contact and reports anomalies to the SVO and SAO.

2.3.4.4 Satellite Analysis Officer (SAO). The SAO is responsible for the performance of the navigation and timing subsystems. This position monitors mission performance using data from the Monitor Stations and through satellite telemetry received via the GAs. The SAO also manages the navigation message generation process, and assists the SSO in uploading the message to the SVs. The SAO provides the primary technical support for navigation mission anomaly resolution for the crew.

2.3.4.5 Satellite Vehicle Officer (SVO). The SVO functions are similar to the SAO, except that this position is responsible for all satellite systems other than the navigation subsystems. These include subsystems such as command/telemetry processors, transmitters/receivers, attitude control, thermal control, and power generation and conditioning. The SVO plans the satellite commanding performed by the SSO. This position also assists in satellite "bus" anomaly resolution.

2.3.4.6 Ground Systems Operator (GSO). The GSO configures and monitors the performance of the GPS communications network and ground stations. The GSO will also perform fault isolation procedures and assist maintenance personnel at the remote sites to correct deficiencies (5:37).

2.3.5 MCS Operations. The MCS is not a general use system; it is specifically designed to perform the task of operating the GPS. As such, there are a finite number of well-defined tasks that the MCS as a system can and does perform. This section describes the MCS task generation and scheduling process.

2.3.5.1 MCS Activities. The GPS program requirement documents are the ultimate source of GPS activities. They specify the eventual end product (a highly accurate radionavigation system) in detail, defining system accuracy, operational rates, availability standards, etc. They describe the goal.

More detailed, "tactical" documents describe *how* the requirements are to be met, in terms of when specific activities are to be performed, who is to do them, and the criteria for success. These documents include manufacturers' technical handbooks, middle echelon requirements documents such as the On-Orbit Maintenance Requirements (OOMR), and Headquarters/Wing/Squadron Operational Instructions (OI). These in turn are broken down into detailed "how-to" documentation for the personnel performing the activities. The Navstar Satellite Support Requirements (NSSR) is one of these, produced by the 2 SOPS to define the timing requirements for each activity performed on the GPS satellites.

For the purposes of this document, and when referring to MCS operations, a *requirement* is the rationale for one or more MCS *activities* to be performed. The requirement originates from the GPS system requirements documents or from operational squadron operating instructions. An *activity* is a single, discrete function, performed by the MCS crew as a whole, that satisfies a requirement.

Table 2.3 shows three of the 78 different MCS activities described in the NSSR. ACTIVITY is the acronym used on the MCS Master Contact Schedule. SCHEDULED describes the scheduling interval; these may be periodic (as shown for ADDKEYS), periodic with caveats (like BAT3TEMP), or aperiodic (like BCHGROFF).

Table 2.3. 2 SOPS NSSR Excerpt (1:7)

ACTIVITY:	ADDKEYS (ADDKEYS transmission)
SCHEDULED:	Twice per week (Normally on Sunday and Wednesday if L1 & L2 transmitters are enabled).
ACTIVITY TM:	10 minutes
PURPOSE:	To upload new navigation processor address keys.
REQUIREMENT:	Execute ADKGT Procedure.
WINDOW:	-1 to +1 hour from scheduled support time.
RESCHEDULED:	Accomplished prior to reusing address keys to avoid unauthorized access to the navigation processor.
ACTIVITY:	BAT3TEMP (Battery 3 temperature monitor)
SCHEDULED:	Once per week when the SVE is 105 degrees and decreasing. (For designated satellites only)
ACTIVITY TM:	10 minutes
PURPOSE:	To determine maximum battery 3 temperature.
REQUIREMENT:	Perform Battery 3 Temp Monitor Ops Procedure.
WINDOW:	+/- 5 minutes from scheduled initiation time.
RESCHEDULED:	Must not violate WINDOW to ensure that battery 3 does not exceed 50 degrees Centigrade while connected to the main bus.
ACTIVITY:	BCHGROFF (Battery CHarGeR OFF)
SCHEDULED:	As required by a schedule request
ACTIVITY TM:	5 minutes
PURPOSE:	To turn a battery charger off
REQUIREMENT:	Perform Battery Charger Off Ops Procedure with BAT.ONE, TWO or THREE.
WINDOW:	-30 to +30 minutes from scheduled initiation time
RESCHEDULED:	Must not violate WINDOW to avoid possible damage to the satellite unless stated otherwise by the schedule request.

The ACTIVITY TM is the expected duration of the activity. PURPOSE and REQUIREMENT assists the scheduler and crew in verifying the need for the activity. The allowed slack about the scheduled time is shown as WINDOW; the requirement may be considered failed if this is not met. Finally, RESCHEDULED contains instructions to the FCMDR on how to reschedule or otherwise react in the case where the scheduled time is missed.

2.3.5.2 MCS Scheduling. Taking advantage of the periodic and well-defined nature of satellite support requirements, routine satellite support activities can be scheduled in advance. Scheduling satellite activities is simply a process of matching the time the activity is due to be performed with the resources needed to perform it. This process is complicated by the shifting SV/GA visibility periods; by scheduled and unscheduled equipment outages; by conflicting requirements; and by additional aperiodic contact requirements that "shift" the time of routine contacts.

The MCS personnel create the GPS Master Contact Schedule every day for the next 24 hour period starting at 0600 GMT. The scheduling process is:

1. Find the visibility of each SV with each GA. The MCS computer system provides this data as a printed report, using the current estimates of SV trajectory to calculate satellite visibility periods and elevations at each of the ground stations.
2. Collect the Schedule Requests (SR) for aperiodic SV contact activities and ground station maintenance. These requests are submitted in writing by staff and support functions to reserve system resources for maintenance and testing at some specific time. The requests are channeled to the scheduling function after approval by the FCMDR.
3. Identify the periodic activities that must be performed on the day being scheduled. This is done by examining the system logs to find the last time these activities were performed, then adding the maximum time interval (from the NSSR). The activity must be performed again prior to the resulting time.
4. Draft the schedule, starting with the SRs with fixed activity start times, then periodic activities. As each activity is added, it must be examined to ensure it meets visibility requirements, meets system requirements, does not conflict with previously scheduled activities, and otherwise has the necessary system resources available to accomplish its task.

2.3.5.3 MCS Operations. During the performance of a satellite support, the MCS crewmembers are cooperating to perform the activities on the Master Contact Schedule. There may be zero, one, or more activities being performed simultaneously in the MCS at any given time, but generally a single crewmember can be involved in only one at a time.

As an example, a common set of activities performed during a satellite contact is a SV state-of-health (SOH) review, a download of SV Global Burst Detector (GBD) processor data (a GBDDUMP), and finally a navigation data upload (NAV). The SSO and the GSO initiate the contact by ensuring the GA is ready. Then with FCMDR authorization, the SSO commands the GA to begin transmitting. After contact is established and telemetry from the SV received, the SSO and the SVO review the status of the satellite's subsystems (the SOH). The SSO then commands the SV to download GBD data (the GBDDUMP). While this was occurring, the SAO was commanding the MCS computer system to calculate "fresh" estimates of the SV's clock and trajectory parameters, form them into a navigation data upload, and transmit the upload to the GA. The SSO then instructs the GA to transmit the data to the SV. The contact ends when the SSO and GSO break the communications link and return the GA to ready status

The schedule has been constructed under the assumption that each of the scheduled activities will take little or no time longer to perform than the activity duration shown in the NSSR. As the result of experience in both scheduling and performing these activities, the durations in the NSSR are conservative and seldom does the performance time exceed the expected. Experienced MCS crew personnel have stated these durations are exceeded as result of satellite, communications, ground station, or MCS computer malfunction (3), (14).

2.4 Modeling.

A model of a system is a representation of the objects, concepts, or ideas of the system in some form other than the system itself (15:2). System models of all types are simply tools for describing system functionality in a language understood by the model builder. Models are also a substitute for real or conceivable systems in cases where the system itself cannot be used for analysis or is otherwise unavailable. That may be due to any number of reasons. The system may not exist (yet), or it may be too expensive or dangerous or inconvenient to test directly. In any case, the situation requiring the model is this: the analyst requires information about the operation of a system that cannot be readily obtained from the system itself.

The environment in which the model is defined is best chosen to optimize the desired result of the analysis. For instance, if the intent of aircraft design research is to evaluate the ability of a particular design to fly, a physical model or prototype can provide this information directly. However, other factors also must be considered. The type of system, the cost of the modeling process, and even the predilection of the researcher also drive the choice of modeling discipline.

However, the explosion of computer technology in the last thirty years has allowed simulation modeling to become the principal method of describing large and complex systems (16:7). Dedicated high-level simulation languages have promoted this growth, as has the inherent flexibility and speed of digital computers.

2.4.1 Model Types. When a model is required, a fundamental decision to be made is what type of model will be created. There is a wide spectrum of model types, each with characteristic strengths and weaknesses. A common classification technique for model types is to place them on a continuous scale according to how they relate to the real system being modeled. Closest to reality are physical and scaled models. Physical models include full-scale mockups and prototype systems. Scaled models are common for physically large systems, such as a world globe modeling the Earth (15:8).

Less concrete are analog models, which have the same functionality as the real system being modeled, but are less like it in form. A graph is a simple analog model; the length of a line or bar on the graph is an analog for some physical property, such as time or mass. Further into the abstract is the computer simulation, where the representation of the real object or system exists only in computer source code and binary numbers. The least concrete of the modeling environments is the mathematical model, where the representation of the system is described only in mathematical or logical symbols. While this level of abstraction is certainly inexpensive and flexible, often the assumptions needed to fit reality into this format reduces the accuracy and usefulness of the model (15:10).

Another consideration when choosing the type of model is the need for stochastic properties. Most models of real-world systems involve some uncertainty in either their input factors or their internal processes. A decision must be made to either include these stochastic properties in the model or to assume reasonable deterministic values and avoid the issue. The model environment must be selected to accommodate either decision (4:13).

2.4.2 Simulation. According to Shannon (15:2), simulation is

> ...the process of designing a model of a real system and conducting experiments with the model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system.

Most researchers recommend a systematic approach to developing a simulation model, and there are frequent commonalities between the approaches. Emshoff and Sisson (4:50) describe an approach shown in flow chart form in Figure 2.6. This sequence differs from that of other authors such as Pritsker (11:10) and Shannon (15:23) in that it emphasizes the *iterative* nature of the model-building process and stresses the creation of subprocesses. This incremental approach is useful for large or logically complex models.

As noted above, the computer simulation is a common method of modeling. It is especially useful when the system being modeled involves stochastic processes or inputs; with the proper programming, the computer can quickly create random variables with many different distributions. The major drawback of computer simulation is its high level of abstraction. In addition, the translation of the desired model into the language of the computer requires specialized knowledge that generally has nothing to do with the environment of the system being modeled.

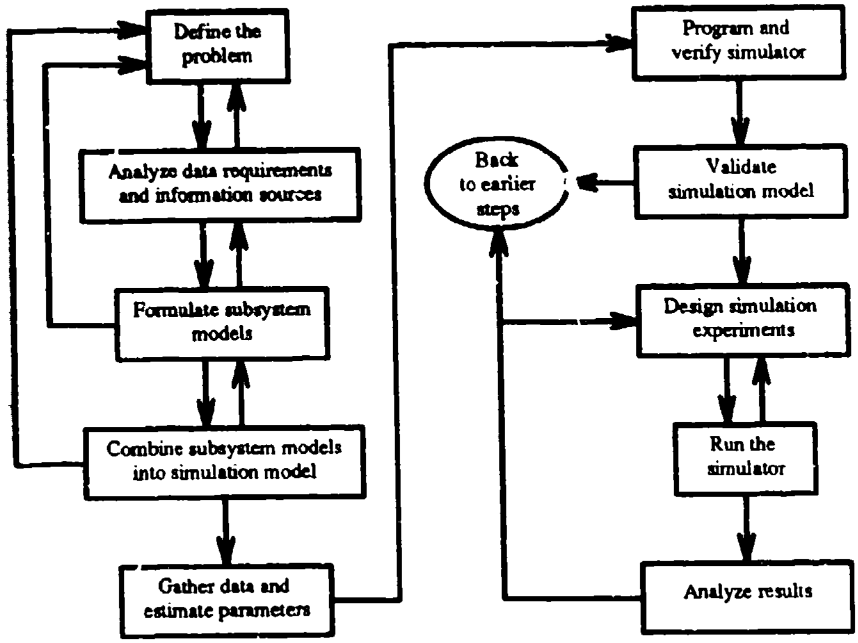
Define the problem
Analyze data requirements and information sources
Formulate subsystem models
Combine subsystem models into simulation model
Gather data and estimate parameters
Program and verify simulator
Back to earlier steps
Validate simulation model
Design simulation experiments
Run the simulator
Analyze results

Figure 2.6. Simulation Model Development Sequence (from (4:50)

2.4.3 Operations Simulation Models. The majority of the research in this area concentrates on the modeling of manufacturing or service systems. Manufacturing systems models have little in common with the operational control systems, as the human element and its unique qualities (dynamic decision-making, reaction times, the capacity to make outrageous errors) so integral in operations is not a major factor in mechanized manufacturing processes.

However, recent manufacturing and service system models give an indication of the "state-of-the-art" of modeling in general. This may lead to the discovery of techniques applicable to operations modeling. Medical facilities, for instance, have a complex operation with unique constraints, and often closely parallels the satellite operations environment. A hospital pharmacy in Knoxville, TN was modeled using SIMAN, a modern personal computer-based simulation language (8:91). The level of detail was down to simulating telephone calls interrupting the pharmacy employees in their duties (8:97). Another study modeled the patient flow in a hospital outpatient center then under construction, and its conclusions resulted in the redesign of the facility (17:26).

Research in modeling the operational environment is generally directed toward the creation and testing of so-called "expert systems". These are computerized decision-making assistants, that have been "trained" by analyzing the optimum decisions made by the human operators and creating a computer program to model this decision-making process. The operators then enter the current situation into the computer expert and this program provides the necessary decision (or decision-making aid) more quickly and accurately than a typical human operator.

The most significant current progress in the design of operational control systems is the creation of the Georgia Tech-Multisatellite Operations Control Center (GT-MSOCC). This facility, a joint project of Georgia Tech University and the National Aeronautics and Space Administration (NASA), is a "...real-time, discrete event, interactive simulator of the operator interface to a NASA ground control system" (13:627). This complex model is used to test new control procedures and mission subsystems for NASA prior to real-world implementation. It is the key facility for the research into human/computer interaction and decision-making process for complex systems (13:636).

2.5 Scheduling.

A major task of satellite operations centers such as the MCS is to optimize the use of the resources needed to accomplish their mission. As described above, the availability of these resources constrain the operations center's ability to perform required tasks in the required time frame. The problem then is to find the mutually optimum time to perform a finite number of tasks of fixed but nonuniform duration, which require some limited set of resources. This problem resembles those categorized in the operations research discipline as "scheduling" or "sequencing" problems. The basic theory of this class of problem is described below, along with common solution techniques and problem examples similar to those found in satellite operations.

2.6 *Scheduling Theory.*

One of the basic problems in domain of operations research is the sequencing, or *job-shop*, problem. This class of problem arises when the optimum order of tasks is to be determined, with each task requiring some fixed amount of each of a number of resources. The archetypical example (and the source of the problem class's name) is a metalworking shop. Each item of finished product must spend a fixed amount of time (dependent of the type of item) on a variety of machine tools, such as lathes, drills, punches, etc. The *order* in which the items are sent to the various machines is not fixed. This case is shown graphically in Figure 2.7(a). The problem then is to find the order or sequence of item/machine pairings that optimizes some specific performance measure. Examples of typical measures are total production time (*makespan*) for one or all items; machine tool idle time; and total waiting time for some number of items to be produced (12:48).

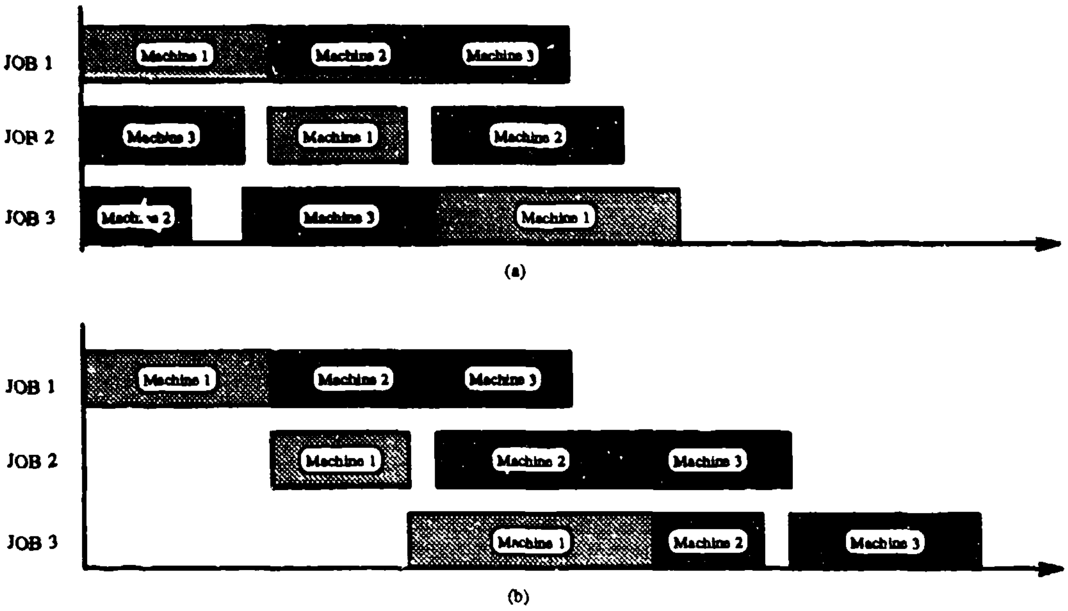

Figure 2.7. Job Shop/Flow Shop Schedules (from (2:180))

If the process requires each item to be smoothed, drilled, and punched an identical order. it becomes a special case of the job shop problem and is called the *flow-shop* type problem (Figure 2.7,(b)). A further subtlety is whether items may skip ahead of others while each item maintains its proper machine sequence. This is called *passing* and naturally complicates analysis of flow-shop problems (12:49).

There are a number of criteria used for judging the effectiveness of schedules. The most common is the amount of time needed to complete all jobs in the system. It is generally desirable to create a schedule to minimize this time, so the criteria is called *min-makespan*. Another type of measure is the mean job flowtime, the average amount of time jobs spend in the system. This is a good indication of the congestion in the system. When the jobs in a system must meet specific due dates, a measure of scheduling effectiveness is *tardiness*. This is the number of jobs whose actual completion time is after the required completion time. The mean or total time in excess of the required completion times for the system can also be used (2:223).

An extension of the simple job-shop problem is the parallel identical processor/independent job problem. The resource uniformity allows simple heuristics to solve this class of problem for many scheduling measurement criteria. Iterative techniques using simulation with some form of priority rules are commonly used to solve these types of scheduling problems. Analytical solutions for problems with two and three machines have been discovered. The common solution method for larger problems is to use a multistage decision process, examples of which are dynamic programming, branch-and-bound methods, and backtrack programming. Each of these is a form of implicit enumeration, that avoids complete enumeration of all existing schedule possibilities (12:49).

III. Methodology

3.1 Overview.

The creation of the MCS Operations Center model follows the strategy described in Section 2.4.2 and shown in Figure 2.6 (Page 2-20). Information describing the processes to be modeled and their quantitative output must be collected. Using these data, measures of performance must be determined, in order to validate the model's performance and measure the experimental result of the model's operation. Once the function, purpose, and measure of performance are known, the model subsystems can be created, verified, validated, and tested. When the subsystems are complete, they are integrated into the whole. This chapter describes in detail how each of these steps were completed for this thesis.

3.2 Data Collection.

Simulating the MCS scheduling process and the operations center task flow required information describing these processes. System requirement information was needed to identify the mission critical activities and how these constrain schedule creation and operations. Operational data were required to define the activity flow internal to the operations center and to derive the actual activity interval and duration time periods. Lastly, historical records were needed to perform model validation. In addition to the above "hard" data, informal interviews were conducted with current and former operational crewmembers to provide initial face validation of the overall simulation concept. The data described as follows was provided by the 2 SOPS to this researcher during an official visit from 29 June to 3 July 1992.

3.2.1 System Requirements. System/mission requirements data were needed to define the constraints under which the MCS created and performed the daily activity schedule. The overall GPS mission requirements are described at the highest level in the Navstar GPS System Operational Requirements Document (SORD). These general requirements, supplemented by technical documentation provided by the GPS satellite manufacturer (Rockwell International), were used by personnel in the 2 SOPS to create the document called the Navstar Satellite Support Requirements (NSSR) (see Figure 2.3, Page 2-15). The NSSR is the primary reference for operational satellite contact requirements. It is utilized by 2 SOPS personnel for scheduling future satellite contacts and for prioritization of new satellite contacts necessitated by real-time schedule changes. For each routine MCS activity, the NSSR describes:

- whether an activity is periodic or scheduled "as needed",
- how often scheduled activities must be performed,
- the expected activity duration,
- the activity's purpose,
- the time window about the scheduled time in which the activity must be performed, and
- the rules to be followed when rescheduling the activity if it cannot be performed at the scheduled time.

The NSSR provides most of the information needed to create a valid GPS satellite contact schedule, while the SORD puts the MCS requirements in the larger context of the entire Global Positioning System. Both documents were obtained directly from the 2 SOPS.

3.2.2 Operational Data. Information describing day-by-day and minute-to-minute MCS operations was required for two reasons. The first was to provide a detailed description of the action and interaction of the six crew positions during each type of MCS activity. This data allows the simulation to include a functional characterization of each position, so that the effects of changes to that position on total MCS performance can be observed. Second, the stochastic tendencies of the operations center must be determined using operational data. While the scheduling process can use the standard activity performance times described in the requirements documents, the operations simulation requires the observed intervals and durations.

The permanent record of MCS operations in terms of discrete activities is maintained at the MCS in the form of Pass Plans. These are detailed descriptions of the activities performed during each satellite contact, including printouts of system displays at the time the activities were performed. Actual event times can be obtained directly from these displays, and the function of each crew position in the performance of an activity can be determined by matching the time and type of commands executed by that position.

Due to the large number of Pass Plans generated by the MCS, only the past 30 days are maintained at the MCS. Pass Plans for 65 satellite contacts performed on 2-3 June 1992 (J152-153) were collected from the 2 SOPS. The data spans all the 17 satellites then operational and all the satellite contact activities performed on a daily basis.

3.2.3 Historical Data. The operational information described above provides the information needed to structure the MCS model and provide statistical data on the operations process. In addition to this, a case history of MCS performance over some set period of time is required to provide the basis for model validation. If the initial conditions at the beginning of that time period can be duplicated, a valid model should reproduce the actual MCS performance within a predetermined tolerance of error.

Completed MCS activity schedules, together with the crew log maintained by the crew Flight Commander, completely describes the activities performed by the crew. The schedule kept for record is the one the FCMDR annotates with contact results, actual (as opposed to scheduled) contact start time and duration, and contacts moved or added on a contingency basis in response to schedule changes. In addition, scheduled and unscheduled MCS computer and GA outages are recorded on these documents.

The complete set of Navstar GPS Master Contact Schedules and Crew Logs from 31 March to 13 May 1992 (J90 to J134) were obtain from the 2 SOPS.

3.2.4 Crewmember Interviews. In addition to obtaining hard data describing MCS operations, personal and unstructured interviews were conducted with the members of an MCS crew.

The discussions occured with "Bravo" crew; Captain Keith Dale was the Flight Commander. The information obtained, while not directly applicable to the model building process, contributed to the overall direction of this project. The results of the interview is described more fully below.

Discussions with Captain James Serpa, a experienced GPS Flight Commander and crew evaluator for that position, were also used to verify model details and to provide additional "face" validation for the model.

3.2.5 MCS Performance Criteria. A number of operational criteria are used by the 2 SOPS to evaluate the performance of the Global Positioning System. These criteria are:

- satellite vehicle availability,
- Master Control Station availability,
- ground station availability,
- navigational/timing accuracy (as perceived by the system),
- operations personnel error rate, and
- satellite support success/failure rate.

Each of these criteria has significance to the system managers and operators. However, their significance cannot be easily inter-related, because they impact (or may impact) different segments of the system. Also, each could have either serious or negligible impact depending on the relative need of the user community for the system at the time of the occurrence, and how that community is affected. Finally, rating the relative impact of these criteria on GPS mission performance is difficult because of the interaction of the events. For instance, a personnel error may remove a satellite or ground station from service, then navigational accuracy may be affected when a satellite support is missed. Although the SORD and other local directives set the acceptable limits for system availability and navigation accuracy, the operational squadron by its nature has a "zero tolerance" for deviations from perfect performance for all the above measures.

Despite the difficulties described above, an accurate indicator of GPS *operations* performance can be obtained. The key is using measurement criteria that are directly affected by crew performance. One of these is the satellite support failure rate. In the terminology of the 2 SOPS, a scheduled support is called "MISSED" if it could not be performed within 60 minutes after the scheduled time, but is completed prior to the maximum allowable interval as specified in the NSSR. A "FAILED" support could not be performed before the time limit, or could not be performed at all (3). Although not officially monitored by the system managers, this discrete GOOD/MISS/FAIL criteria can be extended to a continuous indicator by tracking *how much* time passes until a MISSED or FAILED support is completed.

An *ex officia* criteria for evaluating the performance of the MCS is ground antenna utilization. As described above (and with the exception of PIKE), the GPS ground antennas are GPS dedicated resources. Idle time cannot be used by any other system, so there is no managerial pressure to maximize utilization. However, when the number of scheduled contacts per unit time is high and the system has less scheduling slack, real-time schedule changes are more difficult to resolve without MISSED or FAILED supports. Currently there are more than sufficient "GA-minutes" to allow flexible real-time rescheduling (as is demonstrated by the low MCS satellite support failure rate), but as the satellite constellation grows or GAs break, this utilization statistic will become more important to system decision makers. Even now, GA utilization is an adequate measure of the amount of slack GA resource available to the system.

The MCS model described below uses both satellite support success statistics and GA utilization as performance measures.

3.3 Model Development.

After MCS data was collected and reasonably sound measurement criteria and indicators of success determined, the preliminary decisions regarding the type and basic structure of the

MCS model could be made. For each prospective type of model, there are a number of tradeoffs and assumptions. These involve both internal (having to do specifically with the structure and content of the model) and external (concerning the development *process*) factors. Although the external factors would appear to be irrelevant to the research, they had considerable impact on the methodology and sequence of model development. Once these decisions were made, and after a review of the problem statement and the goals of the project, development began on the model itself.

3.3.1 Goals. In keeping with the tentative solution as described in Chapter 1, the overall goals of the model development process were that the model would:

1. Utilize actual MCS-derived data for initialization and operation.
2. Reproduce the scheduling and operations performance of the MCS with maximum fidelity, given the fact that perfect fidelity is not possible due to the non-deterministic nature of the scheduling process.
3. Provide performance measures that are similar to the mission effectiveness criteria used by the MCS management.
4. Accommodate any set of initialization parameters that were consistent with some past or future state of the Global Positioning System.
5. Be sufficiently clear in construction and operation so that maintenance, follow-on research, or operational use could be performed with minimum effort.

3.3.2 Model Selection. Among the types of simulation models briefly described in Chapter 2, the computer simulation was chosen to be the most apt. The first reason for this selection was the desire to model stochastic activity durations. The two simulation languages available for this research (Simulation Language for Alternative Modeling (SLAM) and SIMSCRIPT II.5) allowed accurate modeling of this facet of the operational environment.

Another reason for the choice of computer simulation was the lack of clear analytical solution methodology for the scheduling problem presented by the MCS. The scheduling technique used, an iterative approach with sliding time windows and changing priorities, was selected because it corresponded to the actual MCS scheduling process. This algorithm was not easily modeled mathematically, although an integer programming or branch-and-bound technique might have been theoretically possible. Even if the scheduling process could have been performed analytically, there would be interface problems with the operations segment of the simulation. The scheduler would have to be called real-time when the operations simulation required a new schedule. The external scheduler would then have to read the current state of the simulation and then provide the new schedule in a form usable to the simulation. These difficulties precluded the use of a "mathematically"-based scheduler model and led to the development of the scheduling "simulator" described below.

3.4 Model Description.

3.4.1 Overview. The following is a detailed description of the MCS simulation code. The source code itself is provided in Appendix A). As the MCS.SIM source code is well documented, this description will concentrate on fitting the individual routines into the larger structure of the program, and will describe the rationale and logical basis for the techniques used. This description assumes some knowledge of the SIMSCRIPT II.5 programming language; readers are referred to the language manual (Bibliography reference (7)) for answers to syntax questions.

Coding conventions were standardized and applied consistently throughout the program to allow easier understanding and modification. These conventions are.

- A block of asterisks separate SIMSCRIPT routines and processes,
- Statements affected by conditional (IF) and flow control (DO...LOOP) constructs are indented from the controlling statement for every level of nesting,
- Remarks *precede* the sections of code they describe and are inset to the appropriate level,
- Local variable names start with a period(.),
- Variables and entities are UPPERCASE.

3.4.2 Initialization Process. This section describes the sequence of events necessary to initialize the simulation. The initial conditions for the simulation run are maintained in data files external to the program. This allows different simulation runs to be made without modifying the simulation source code. These initial conditions include data that describes the MCS state at the desired simulation start time, the desired resource changes that occur *during* the simulation, and operational parameters that define the limits of the simulation run. The routines described will be the PREAMBLE and the READ.DATA, READ.VIS, INIT.GA.USE, and INIT.ACTS routines.

Reader familiarity with the following frequently used program variables and entity attributes is helpful during the discussion of the program's operation. In some cases, the same attribute names are used with more than one entity. "SV" and "GA" are text attributes that store the names of satellite vehicles and ground antennas. The SV names are each seven characters, starting with either "BII-" or "BI-" (Block I or II), followed by a number indicating the launch order. Satellite numbers greater than 20 indicate the satellites not yet operational at the time data was collected on the system. "GA Index" is an integer assigned to each GA to simplify the manipulation and selection of ground antenna data.

3.4.2.1 PREAMBLE Section. As with all SIMSCRIPT programs, the PREAMBLE section is used primarily to define variables (7:45). System resources, entities, queues, and processes are also defined and their attributes described. Of note in the PREAMBLE are DEFINE statements that set the rank ordering of the queues used by the scheduling and operations routines. These commands play a critical role in these processes and will be explained in detail in later sections. The TALLY statements create monitoring subroutines that continuously collect statistics on system variables relating to simulation performance (7:240).

3.4.2.2 READ.DATA Routine. This routine opens the default input data file, of the name specified by the user at program execution time. This file contains the filenames of the remainder of the input and output files. The first file in the list (with filename assigned to the text variable TIME.FILE) is then opened. The desired simulation start and end times (in 1992 Julian day/hours/minutes) are then read, converted to the number of minutes since 0000 hrs on 1 January 1992 (henceforth called "jminutes"), and then stored in the variables SIM.START.TIME and SIM.END.TIME. The variable CURRENT.TIME is set to SIM.START.TIME; this will not change until after the initial scheduling is performed and the simulation clock starts. The next values read from the file are three simulation parameters. NUM.SIMUL.CONTACTS is the maximum number of simultaneous satellite contacts that can be performed by the MCS during the simulation. Technically, the most the sys.em can handle is four (5:7) . However, because there are currently only three SSO's per operational crew, the practical limit is three. The variables MAX.PRIORITY and INTERVAL.OFFSET.STEP are used by the scheduler in managing the rescheduling process. Finally, this routine creates the system resources used in the management of the operations simulation. Refer to Point A on Figure 3.1, which is the first of a number of figures designed to show the operation of thesimulation schematically. Each labeled block represents a data storage structure, either an array or a queue. In the case of queues, the data are stored as SIMSCRIPT entities. The simulation functions primarily by moving these entities among the various queues at the appropriate times. The order in which the entities are maintained and accessed is also important. As each entity has a start time attribute, the entities are generally in forward or reverse chronological order.

3.4.2.3 READ.VIS Routine The READ.VIS routine opens the file indicated by the VIS.FILE text variable and reads data that describe the time intervals when each GPS satellite used in the simulation is visible at each of the selected ground antennas. Each period of visibility is then stored as a temporary entity and placed in a queue called VIS.TABLE. Refer to Point B on Figure 3.1.

Any source of visibility data can be used to provide data for this program. Initially this routine contained the code necessary to create the satellite visibility tables using ground station

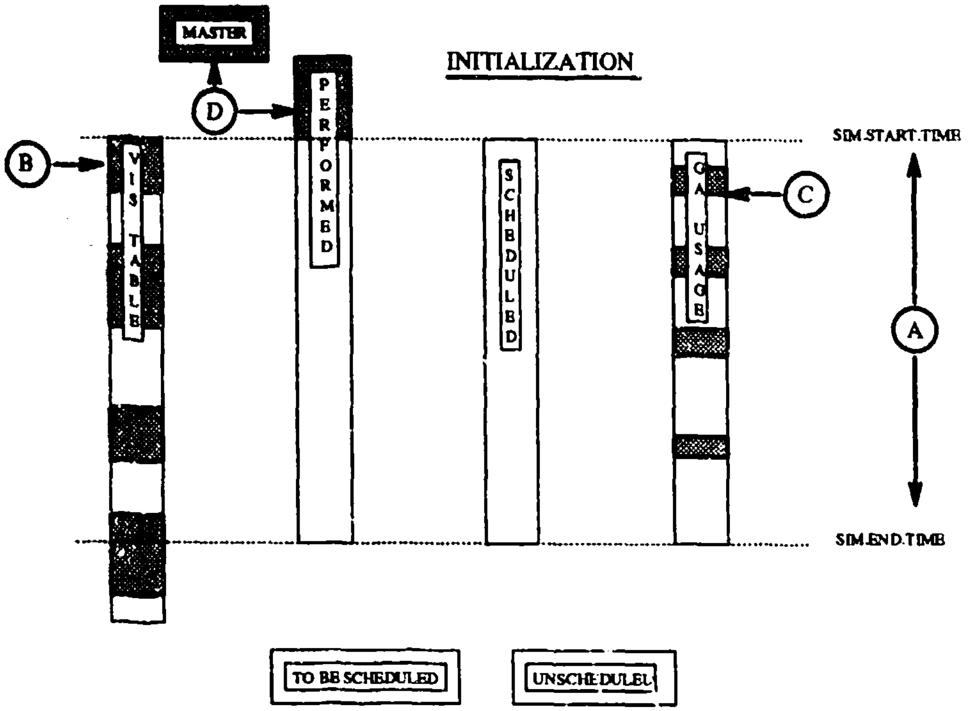

Figure 3.1. Simulation Initialization

locations and satellite orbit element sets. However, the complex and repetitive calculations required by this method proved to be time-consuming to run and difficult to code in SIMSCRIPT. The practical alternative was to create the data externally. The optimum source of visibility data is the MCS-produced visibility charts, especially when attempting to duplicate past MCS performance. However, this data was not available for the time period used by the simulation, so this research used the data produced by Pass Scheduler, written in Pascal for MS-DOS-compatible computers by Kelso (8). This program, using standard two-line orbits element sets and the latitude, longitude, and elevation of the viewing location, calculates satellite rise and set times for a specified time period. The results can be saved as a text file and easily manipulated to conform to the input specifications of the READ.VIS routine (described below). The program Pass Scheduler and its use with this research is described in detail in Appendix D.

The first line of the READ.VIS data file is the number of ground antennas used by the simulation. The second line contains the number of days from the start of the year to the start of the month being simulated, including 1 January and excluding the first day of the month. The visibility data is then read one line at a time, with each line containing the data for one SV/GA visibility period. Each GA list, which can contain any number of satellite visibility periods for each satellite, is structured as follows (values are separated by one or more space):

- Line 1:
  - Name of the GA,
  - GA Number.
- Line 2 through n:
  - SV name,
  - Start visibility day-of-the-month,
  - Start visibility hour,
  - Start visibility minute,
  - End visibility hour,
  - End visibility minute,

After the final entry for each set of Ground Antenna visibility entries,the word "END" followed by five zeroes separated by spaces end the list for that GA; the next GA set (if any) starts on the next line.

For each line of data containing visibility data read, a VIS.EVNT entity is created. This entity has five attributes: Rise time, Set time, SV name, GA name, and GA index. The rise and set times are stored in jminutes.

3.4.2.4 INIT.GA.USE Routine. The purpose of this routine is to initialize the preplanned times when some or all of the GPS Ground Antennas will be unavailable for use during the simulation. During the validation stage, this capability allows the simulation to model the historical MCS and GA outages. For experiments, the outage data provides a significant test variable for evaluating system response. MCS total-system outages can be simulated by making all Ground Antennas unavailable for the duration of the outage. Refer to Point C on Figure 3.1.

The desired outage times are stored in the datafile pointed to by the OUTAGE.FILE text variable. The first two lines of the file are text strings. These strings allow the user to describe the contents or use of the file; the strings are reprinted at the end of the simulation along with the results of the simulation. The next line is an integer value indicating the number of outage description lines to follow. Each line contains (separated by spaces):

- the GA index number,
- the GA outage start time in jminutes,
- the outage duration in minutes,
- the reason for the outage, in one word.

To store the above data, a two-dimensional matrix called GA.USAGE is created. The first index is the GA index number; the second is the simulation time, in minutes since simulation start. Each entry in the matrix is a text value (a "string" of characters) describing how each GA is being used at each specific minute. These strings replace the null strings (text variables containing no characters) which are the default matrix entries. For each minute the GA is planned to undergo maintenance or be otherwise unavailable, the value of the matrix cell for that minute and that GA will contain the one-word description of the outage. Unreserved minutes will contain a string of ten space characters, which are better for output formatting than null strings.

A number of techniques were considered to maintain the GA use record, including an entity/attribute list similar to the VIS.TABLE and the use of system resources to manage GA utilization. The large number of table look-ups required made the entity method too slow, while the large number of resource entities required (one for each GA-minute for the duration of the simulation) stretched the storage capacity of the VAX-implemented SIMSCRIPT software. The matrix scheme described above is the result of a compromise between processing speed, storage capacity, and programming convenience.

3.4.2.5 INIT.ACTS Routine. In order for the simulation scheduling routine to schedule new MCS activities conforming to system requirements, information on previously performed activities must be known. In this simulation, each distinct task performed during a contact for a specific satellite is called an "ACTIVITY", or "ACT". Each ACT is a SIMSCRIPT entity. The details of the ACT required for the scheduling and operations functions are maintained as attributes of the ACT entity. This routine reads contact history data contained in the file pointed to by the VIS.FILE variable. The data describes the name, start time and other details of the most recent (prior to the simulation start time) occurence of each type of contact activity. For instance, only the description of the most recent navigation upload activity for satellite BII-010 would be included in the input data file. Refer to Point D on Figure 3.1.

After opening the specified file for input, the routine reads first the number of satellites being simulated. then the number of individual activities to be read. A permanent entity called a MASTER.ACT is created for each of the activities to be read; the input data is then assigned to attributes of these entities. The activity attributes fall into two categories: those that remain unchanged during the life of the entity, and those that are modified during the simulation. The fixed attributes are:

- NAME: Text label describing the function of the activity. For the basic simulation, the labels are:
  - NAV - Navigation data upload,
  - SOH - Satellite state-of-health determination, using the satellite telemetry data,
  - NAVBUFF - Navigation processor data buffer dump,
  - GBDDUMP - Global Burst Detector processor data dump,
  - NDUTLM - Navigation processor real-time telemetry examination.
- SV: Text label indicating the satellite vehicle pertaining to this activity. The label consists of the the Block number, a dash, then the satellite launch number. Future satellites have launch numbers greater than 20.

- BLOCK: The satellite Block number, either the integer 1, 2, or 3. Research and development vehicles are Block I (1), the first series of operational satellites are Block II (2), and the later version of the Block II SV is labeled (3).
- DURATION: The expected duration of the activity in integer minutes.
- VARIANCE: The variance of the activity duration. This real value is used in the operations simulation process, not in the scheduling process. Defaults to 1.0
- INTERVAL: The maximum interval between occurrences of this activity, in minutes.
- CRITICALITY: This integer value describes the impact of this activity on MCS mission completion. Planned to be used as another tiebreak criteria for scheduling conflict resolution, a system not currently implemented in the simulation code. Defaults to 1.

The dynamic activity attributes are:

- GA, GA.INDEX: After the activity is scheduled or performed, the text attribute GA stores the abbreviated name of the location of the Ground Antenna that did or will perform the contact. The GA.INDEX attribute designates each Ground Antenna with an unique integer number. The values used in the simulation are:
  - ASCN (1)- Ascension Island
  - CAPE (2)- Cape Canaveral
  - DIEG (3)- Diego Garcia Island
  - KWAJ (4)- Kwajalein Island
  - PIKE (5)- Falcon AFB, Colorado (PIKE is the AFSCN designation)
- START.TIME: This is the time of either the scheduled activity performance (if after the current time) or the actual performed time (if prior to current time). The integer value is in jminutes.
- NEXT.START.TIME: This attribute is used to hold the tentatively scheduled start time of the next occurrence of this activity during the scheduling process. The format is the same as START.TIME.
- PRIORITY: This integer number is the primary means for deconflicting activities during the scheduling process. The default is 0, which is incremented if this activity conflicts with another while being scheduled.
- INTERVAL.OFFSET: The scheduling process will allow activities that are difficult to schedule to exceed the maximum time between activity performance, if changing the activity priority alone does not resolve the scheduling conflict. This attribute stores the current value (in minutes) of the amount of excess interval allowed. Default=0.
- LEAD.TIME When an activity is selected to be scheduled, the time remaining from CURRENT.TIME to the next expected START.TIME is calculated (the units are minutes) and stored in LEAD.TIME. This value represents the "urgency" of scheduling the next activity of this type, and can be used to prioritize the activities waiting to be scheduled.
- STATUS: This text attribute describes the current status of this activity. The permissible values are:

- UNSCHD - Unscheduled
- SCHD - Scheduled
- TENT - Tentatively Scheduled
- RESCHD - Reschedule
- MISSED - Activity has exceeded maximum PRIORITY
- FAILED - Activity has exceeded maximum INTERVAL.OFFSET

The newly-created MASTER.ACT entities are then filed in a queue labeled MASTER; these serve as a permanent record of the original activities and as a tool for indexing when temporary entities representing activities must be accessed. A set of temporary entities identical to the MASTER.ACT entities is also created at this time and filed in the queue PERFORMED. These form the starting reference for the scheduling routine, representing the activities performed just prior to the start of the simulation.

3.4.3 Scheduling Process. This section of the MCS simulation code creates a valid MCS schedule for the period of time from the current simulation time to the end of the simulation. The MCS schedule is a chronological list of activities that are to be performed. While this simulation only pertains to satellite contact activities, the real MCS schedule also lists equipment outages for maintenance, system housekeeping activities, scheduled communication requirements, and so on.

For this application, a "valid" schedule is a satellite contact schedule that satisfies MCS operational constraints and either allows all the NSSR satellite contact requirements to be met or flags those requirements not attained. This models operational reality, as the MCS constraints are fixed by the MCS hardware (i.e., number of Ground Antennas) or operational software (i.e. maximum number of satellite the system can operate). On the other hand, the NSSR-specified requirements can often be "stretched"; generally there is no concrete and immediately deleterious effect of exceeding the maximum interval between activities called out by the NSSR. Tradeoffs can (and often must) be made, such as delaying a STATE OF HEALTH activity on a reliable satellite (until after its maximum interval) to allow a contingency NAV UPLOAD activity to correct navigation signal errors. The result is that late or missed activities do not invalidate a schedule, as long as the deviations from the system requirements is known.

In addition to the above requirements, there are other aspects to the schedule creation process that must be considered. The system requirement documents rarely specify the minimum interval between activities, so a valid schedule could be created that would be inefficient due to activities being too frequently performed. So activity intervals should be maximized inside the bounds allowed by system requirements. Another consideration is the question of priority. Given limited satellite contact resources (time, antenna availability, satellite visibility, etc.), there will be instances where satellite activity requirements for one or more satellites cannot be met. Which activity has priority for the available resources?

The scheduling process addresses these issues. Three routines support the scheduling function: MAKE.NEW.ACT, PRESCHEDULE, and SCHEDULE.

3.4.3.1 MAKE.NEW.ACT Routine. The smallest unit of schedulable time in this simulation is represented by the ACT entity. As will be seen below, the presence of this entity in a particular queue is the primary mechanism of the scheduling process. As the reproduction of an ACT entity is a common requirement in the process, it was efficient to create a generic routine to perform this function whenever required.

This routine is passed the pointer of the entity to be duplicated by the calling routine. A new ACT entity is then created, its attributes identical to the original. The pointer to the new entity is then passed back to the calling routine and this routine ends.

3.4.3.2 PRESCHEDULE Routine. Fundamental to the scheduling process is knowing which activities need to be scheduled. This routine selects the activities (if any) that must be performed at least once more prior to the end of the simulation, based on the last time it was performed or scheduled to be performed. If there are activities still to be scheduled, this routine then sets up the system queues to the configuration expected by the SCHEDULE routine and calls that routine. If there are no activities requiring scheduling, it prepares the system for the operations simulation process and exits. The PRESCHEDULE routine uses the activities in the MASTER.ACT queue as means of indexing through every SV/activity pair exactly once. As described above, there is one entry in this queue for each activity required by each SV (for instance, SV BII-010 will have NAV, NAVBUFF, GBDDUMP, and SOH activity entity in MASTER.ACT). The INIT.ACTS routine also placed a copy of each of these entities in the PERFORMED queue, which is rank ordered by latest (i.e., numerically largest) START.TIME. For each activity in turn, the queue PERFORMED is searched for the *latest* occurance of that activity (Point A in Figure 3.2).

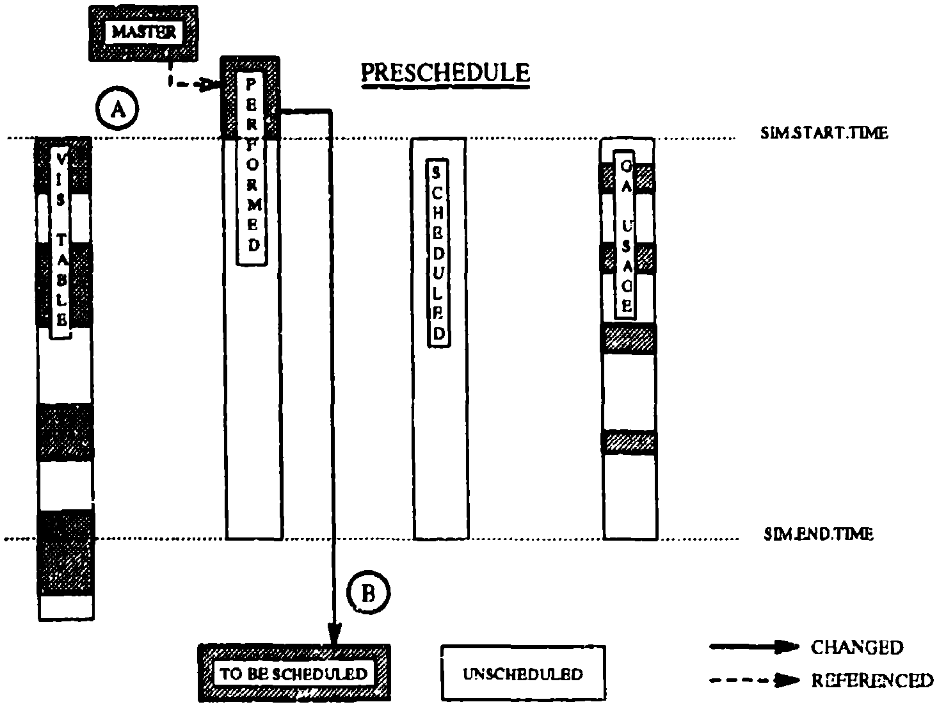

Figure 3.2. Prescheduling Routine

The status of the found entity will depend on when in the simulation PRESCHEDULE has called. At simulation start, the latest ACT entity is one of those loaded at initialization, with a START.TIME prior to the SIM.START.TIME. If the simulation has been running and a new schedule is required, the latest ACT would be one that the operations simulation has just performed, with a START.TIME indicating the simulation time the activity was started. Once the desired ACT entity with latest START.TIME is found, it is evaluated to determine if another activity of this type for this SV is required in the time frame of the simulation. This will be true if the activity interval plus the interval offset plus the activity duration is less than the time remaining from the last occurrence of this activity to the end of the simulation. This is shown graphically in Figure 3.3. In the figure, new activity “A” will be scheduled, while “B”, the same activity but with a larger INTERVAL.OFFSET, would not be scheduled. Adding the activity DURATION to the previous START.TIME prevents the system from scheduling contacts that cannot be completed in the simulation time interval.

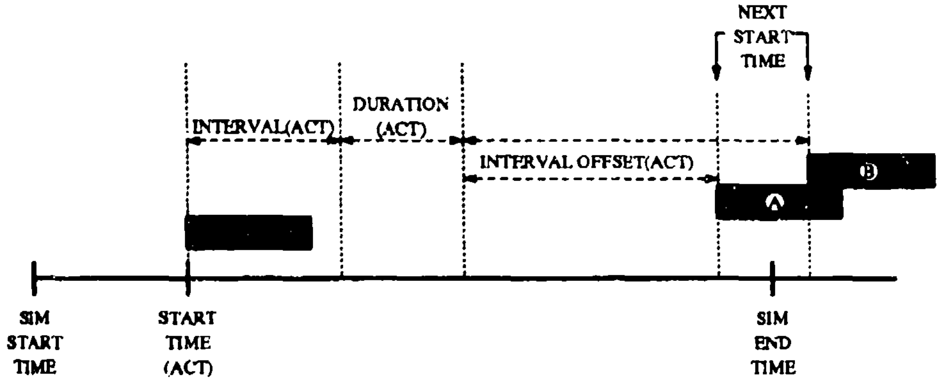

Figure 3.3. Activity Selection Timeline

If the activity must be performed again, a copy of the previous ACT entity is made. This new ACT has its STATUS attribute changed to “UNSCHD” and its LEAD.TIME calculated. The values of PRIORITY and INTERVAL.OFFSET are collected for statistical analysis, then reset to zero. Finally, the new ACT entity is filed in the TO.BE.SCHEDULED queue (Point B in Figure 3.2).

If the above steps have been completed without finding a candidate activity to schedule, the TO.BE.SCHEDULED queue will now be empty (it is always empty when PRESCHEDULE is called). If so, the scheduling process is complete; all the tentatively scheduled activities have their

STATUS changed to "SCHD" and are moved to the SCHEDULED queue. The entities representing "FAILED" supports remain in the SCHEDULED queue, indicating a failed attempt to schedule an activity for the time near the value of START.TIME. If there are entities in TO.BE.SCHEDULE at the end of PRESCHEDULE, the routine SCHEDULE is called to attempt to fit these activities into the tentative schedule.

3.4.3.3 SCHEDULE Routine. Once the activities that require scheduling have been determined, they must be scheduled in accordance with the requirements described in the NSSR and the constraints driven by system resources. In addition, the activities should be scheduled *smartly*, with consideration given to the efficient use of system resources. The scheduling process is basically sequential trial-and-error. The activities to be scheduled are examined one-by-one and the optimum time to next perform that activity is determined. Then, starting at that time and moving toward current time, the system satellite contact resources are scanned for a suitable starting time. At the first instance when all the necessary resources are available to perform the activity, the search process stops. The activity is then tentatively scheduled for that time and resources are reserved.

If any one of the current activities waiting to be scheduled cannot be scheduled in the allowed time window, all activities, both tentatively scheduled and still waiting, are returned to be scheduled again. However, the activity creating the conflict is increased in priority so that it gets the first opportunity to reserve the needed resources. The entire process is then repeated. The next escalation in conflict resolution occurs when, due to repeated conflicts, the priority of any activity has been increased to a predetermined threshold. At that point, the process "gives up" trying to find a spot in the schedule that meets the maximum interval requirement. It now tries to schedule the activity to minimize lateness. The search interval for that activity is expanded slightly and the schedule process repeated. Finally, if the lateness of the activity reaches a set maximum value without the activity being successfully scheduled, the scheduling routine simply stops trying. It inserts a specially-tagged ACT entity to hold the place of the missing activity and continues to schedule without the troublesome activity. The implementation of the above scheme in SIMSCRIPT is described in detail below.

The first step is to remove the first ACT entity with a STATUS of "UNSCHD" from TO.BE. SCHEDULED. Which entity is first depends on the rank ordering for this queue as defined in the PREAMBLE. During simulation validation, the ordering was optimized to minimize the difference between the activity start times as scheduled by this routine and the times the activities were actually scheduled by personnel at the MCS. The closest match occurred when the entities in TO.BE.SCHEDULED were ranked by their high PRIORITY, then low INTERVAL, then low START.TIME attributes. After the entity is removed from TO.BE.SCHEDULED, its attributes are referenced using the entity pointer called ACT for the remainder of the routine. This process is shown as point A on Figure 3.4.

Now the time period over which the "resource space" is searched is determined. This interval begins at START.TIME(ACT) (which is the last time this activity was started for this SV), plus the INTERVAL(ACT), plus the INTERVAL.OFFSET(ACT). This is the *latest* simulation time at which this activity will be tentatively scheduled. Then stepping back to either START.TIME or CURRENT.TIME, whichever is later, each minute is tested as a candidate starting time for this activity. The current candidate minute is stored in the variable .CNCT.START, shown as Point B on Figure 3.4.

Once a possible contact start time is set, each VIS.EVNT entity (which define the GA/SV visibility periods) in VIS.TABLE is tested for: 1) applicability to the SV of this activity, 2) visibility that begins prior to the tentative contact start, and 3) visibility that ends after the duration of the activity. The attributes of the first VIS.EVNT entity to meet these requirements (see point C on Figure 3.4) are referenced in the remainder of the routine by the pointer VIS.EVNT. Most

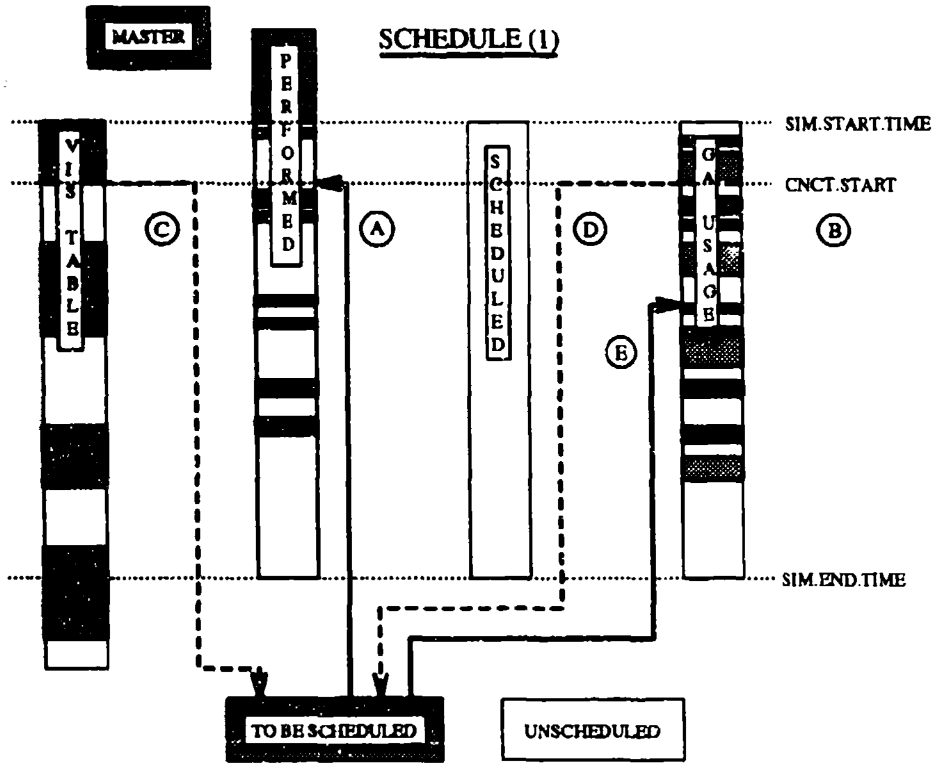

Figure 3.4. Scheduling Process (1)

importantly, the VIS.EVNT selected determines the GA (name) and GA.INDEX at which the tentative schedule entry will be performed.

The last check to be made is to ensure the Ground Antenna referred to by the VIS.EVNT is not reserved by some other activity or scheduled event (Figure 3.4, point D). The status of each GA is maintained in the GA.USAGE two-dimensional text array. The first index the GA number and the second a count of the simulation time in minutes. The minute index of the array runs from 1 to SIM.END.TIME - SIM.START.TIME + 1, so a correction is needed to convert simulation time to the array index. The array index, .OFFSET, corresponding to the simulation minute being examined is equal to .CNCT.START+.t-(SIM.START.TIME-1). One must be subtracted from SIM.START.TIME to correct for the fact that the simulation starts at time zero but the GA.USAGE array starts with index "1". To test each minute from the prospective contact start time through the duration of the contact, the counter ".t" is incremented from zero to the DURATION of the ACT, as long as the .CONFLICT flag, earlier set to zero, stays unchanged. For each minute, the following occurs:

1. The target GA is tested for previous use,
2. Every GA is tested for a previous reservation for the SV being scheduled,
3. The total number of GAs reserved for satellite contacts is tested for exceeding the number of simultaneous contacts (NUM.SIMUL.CONTACTS).

If any of the tests are true, the .CONFLICT flag is set and the .t loop ends immediately. This condition results in the program flow bypassing the "no conflict" structure (described below). The next VIS.EVNT in the VIS.TABLE that describes favorable SV/GA visibility (if any) is referenced. If there is dual (or triple) visibility, the program again examines GA.USAGE for those other GAs at this simulation time. If all GAs visible to this SV turn out to have no free time for this contact, the VIS.EVNT loop ends, the .CNCT.START counter is decremented, and the entire process repeats for the simulation minute prior to the previous. This process will continue until all possible contact start times have been examined, or until a place is found in the schedule for this activity.

If at any time in this process a suitable place in the schedule is found, the attributes of the ACT entity representing this activity are modified. The lucky .CNCT.START time is stored in NEXT.START.TIME, the GA name and GA.INDEX (from the VIS.EVNT entity) are stored in like-named ACT attributes, and STATUS(ACT) is changed to "TENT". When the STATUS is changed, the do-loops stepping through the .CNCT.START times and the VIS.EVNT list stop indexing and fall though their "loop" statements to the remainder of the routine. Lastly, the SV and NAME attributes from the successfully (but tentatively) scheduled activity are written into the GA.USAGE array for the scheduled times, reserving this resource for this activity (Figure 3.4, point E).

If the process described above completes without finding a place in the schedule for this activity (shown as Figure 3.5, point A), further steps are called for. First, if the current PRIORITY of this entity is less than the maximum, the PRIORITY is incremented by the current value of .PRIORITY.STEP. This variable increases as the number of activities *not* being scheduled increases, which adds variability to the prioritization process. The entity's STATUS is then changed to "RESCHD" (to allow tracking) and the .RESCHD.FLAG is set, which later directs the process to bypass the PRESCHEDULE routine and repeat SCHEDULE with the current but subtlety different list of activities. If this ACT entity's PRIORITY is at or greater than maximum, its STATUS is changed to missed and its INTERVAL.OFFSET is incremented by the value of INCREMENT.OFFSET.STEP. This expands the search range of the schedule process, with the understanding that the maximum interval between activity performances has been exceeded in this case, and the critical action is to schedule the activity as soon as possible.

If after the above operation the INTERVAL.OFFSET value has not exceeded one-half the original INTERVAL, this activity is placed back into the TO.BE. SCHEDULED queue (Figure 3.5, point B) to try to find a place in the schedule the next time SCHEDULE is called. If the maximum interval offset has been exceeded, the entity STATUS is changed to "FAILED" and the FAILED variable is incremented. The activity's START.TIME attribute is set to the optimum simulation time for this activity to be performed, had it been able to be scheduled. Then a copy of this entity is made (using the MAKE.NEW.ACT routine) and stored in the PERFORMED queue. This holds the place for this activity and allows future activities of this type to be scheduled. With a STATUS of "FAILED", this entity will not be moved to the SCHEDULED queue to be performed by the operations simulation process. The original ACT entity is stored in the UNSCHEDULED queue (Figure 3.5, point C), so that it's attributes can be examined during the output phase of the program.

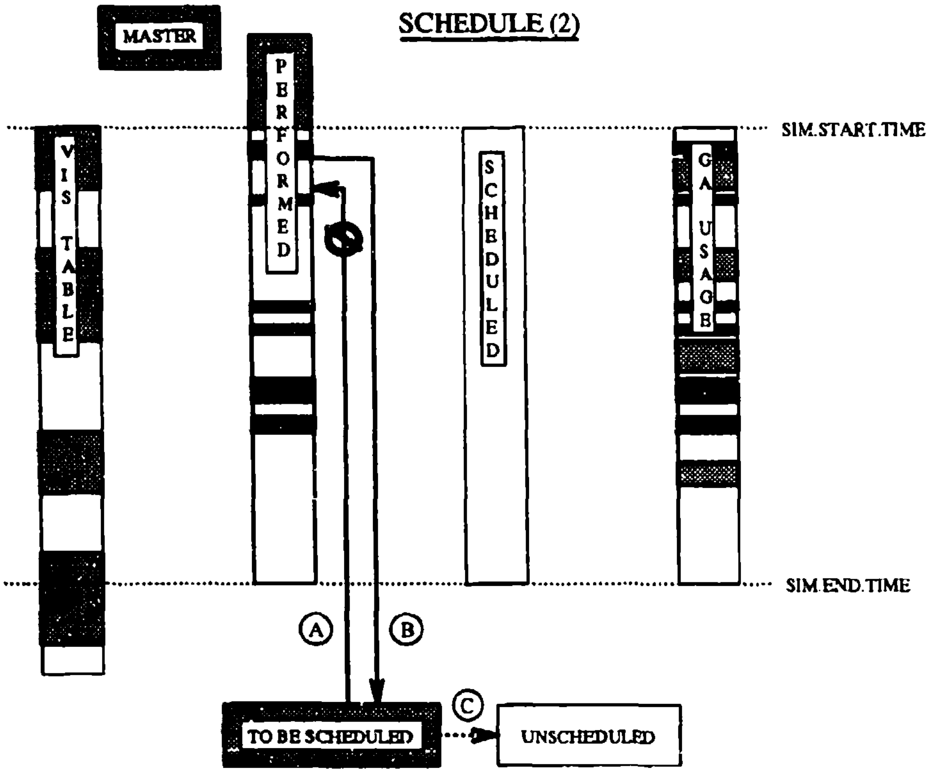

Figure 3.5. Scheduling Process (2)

If the above process completes without scheduling conflicts for any reason, the tentatively scheduled activities in the TO.BE SCHEDULED queue have their new start time (stored in NEXT.START.TIME) copied to the START.TIME attribute. The STATUS is set to "TENT" and the entity moved to the PERFORMED queue, where it becomes the basis for planning the next batch of new activities. Now the TO.BE SCHEDULED queue is empty, the new set of activities are in PERFORMED (but marked as tentative), and the PRESCHEDULE routine is called to determine whether further scheduling in necessary. As noted in Section 3.4.3.2, if no further activities require new contacts, the tentatively scheduled ACT entities in PERFORMED are all promoted to "SCHD" STATUS and moved to the SCHEDULED queue for input to the operations simulation segment (Figure 3.6).

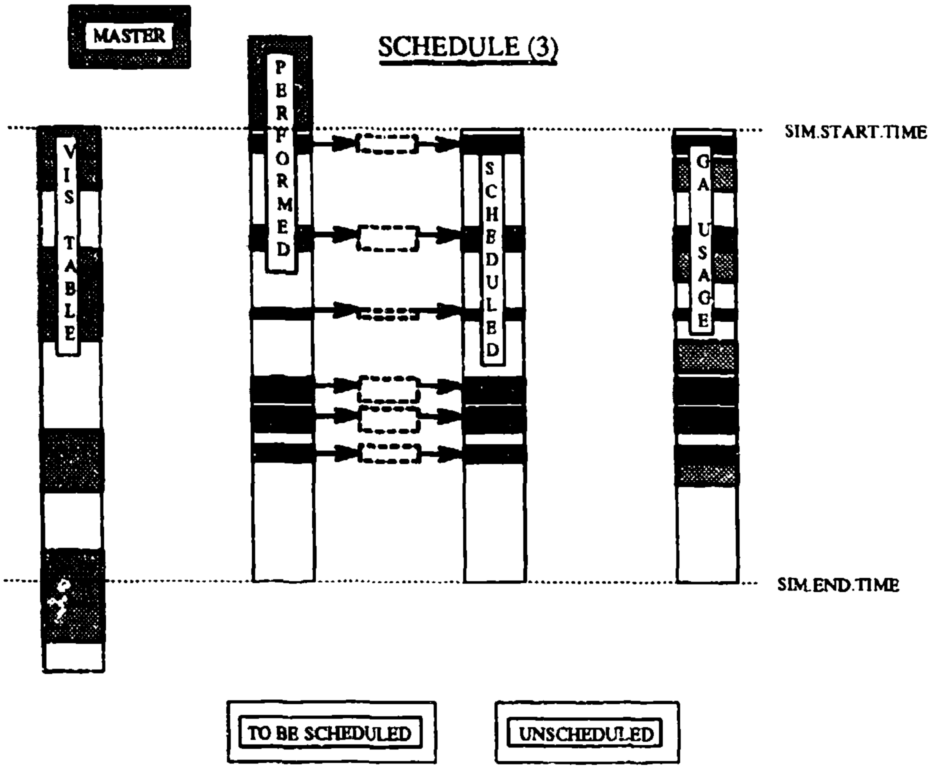

Figure 3.6. Scheduling Process (3)

On the other hand, if scheduling conflict did occur, the attributes of the tentatively scheduled a tivities in TO.BE.SCHEDULED are used to erase their GA reservations in the GA.USAGE array. They then are marked as unscheduled and the system is back to the state it was in when SCHEDULE was first called, except that the conflicting activities either: 1) have new PRIORITY and therefore will be scheduled in a different order, 2) have a larger INTERVAL.OFFSET, allowing a longer simulation time period to be searched, or 3) have been marked as "FAILED" and removed from scheduling consideration. At this point SCHEDULE is called again and the "old" entities in TO.BE.SCHEDULED try again for space on the schedule.

3.4.4 Operations Simulator The task of the operations simulation segment of the MCS Simulator is simply to perform the MCS schedule as devised by the scheduling process. Given the complexity of the activities performed and system performing them, and the fact that human beings are "in the loop", some variability in the performance of the scheduled activities is to be expected. The operations simulation, in attempting to emulate the function of the MCS, adds that uncertainty. The function is performed by two processes: START.OPS and PERFORM.ACT.

3.4.4.1 START.OPS Process. This segment of the MCS Simulation consists of two processes. The first, called START.OPS, takes no simulation time to perform its function and occurs once every simulation minute. START.OPS first updates the variable CURRENT.TIME by adding the time from simulation start to SIM.START.TIME. It then checks for available CONTACT resources; there will be NUM.SIMUL.CONTACTS (provided by the user at initialization) of these resources. If one or more CONTACTs are available, the next scheduled activity is selected (Point A on Figure 3.7) and handed off to the PERFORM.ACT process (described below). This next scheduled activity is the ACT entity in the queue SCHEDULED with the lowest START.TIME. If the simulation end time has not been reached, this process then schedules itself to occur again in one simulation minute.

3.4.4.2 START.OPS Process. Once an activity has been selected to be performed, the PERFORM.ACT process simulates the execution of that activity. The first step is to reserve one of the CONTACT resources for the duration of the process. The actual duration is then determined. If statistical deviation from the standard duration for this activity is desired, the DURATION and VARIANCE attributes of this ACT entity are used in conjunction with internal SIMSCRIPT statistical distribution functions to calculate a new activity duration. This value is maintained locally in the .DURATION variable, and is also written back into the DURATION attribute of this entity for "permanent" record.

Next, the scheduled start time of this activity is compared to the CURRENT. TIME. If the START.TIME(ACT) is greater the the current time, the process waits until the activity start time to continue. If the activity scheduled start time has passed, the process checks the VIS.TABLE

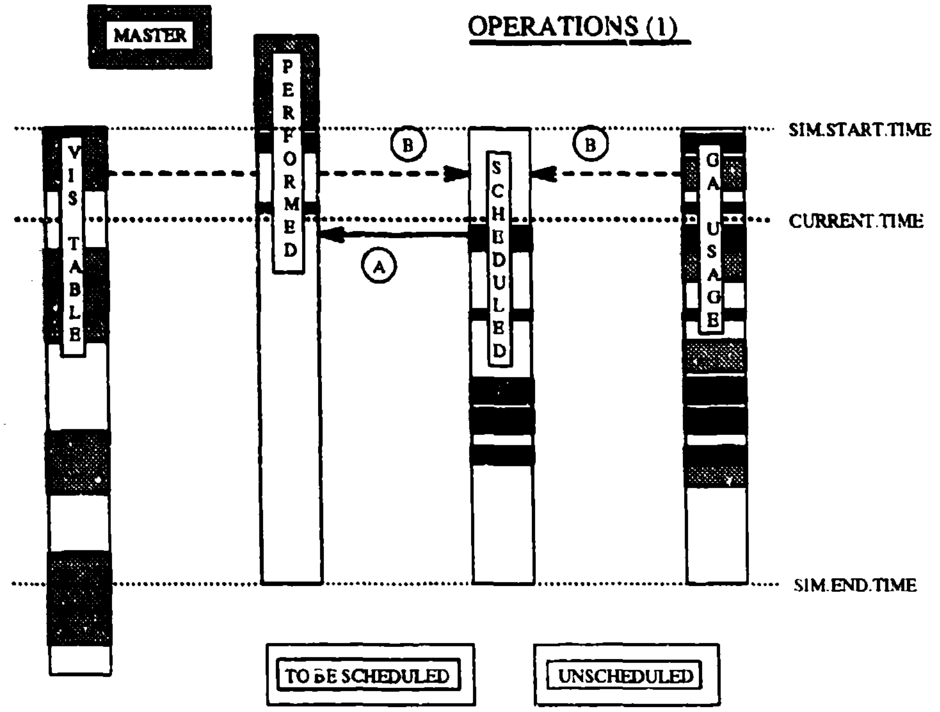

Figure 3.7. Operations Process (1)

queue and GA.USAGE array for sufficient visibility and free time at the Ground Antenna scheduled for this activity (labeled B on Figure 3.7). If enough visibility and GA time is available to perform the activity beginning at the current time, then the process continues and the scheduled activity is performed starting at the current time. If the activity cannot be performed now due to lack of GA visibility or free time, .OFF.SCHEDULE is set to 1.

If this activity can be performed at this CURRENT.TIME, the ACT entity start time is changed to the current time, the STATUS is changed to "PERF", and the process waits the number of minutes specified by the DURATION attribute. After the wait is completed (DURATION minutes later), the process returns the CONTACT resource to the common pool and files this ACT entity in the PERFORMED queue. If the .OFF.SCHEDULE flag is set, the current activity cannot be performed at this time, and the current schedule is no longer valid. A new schedule starting at the current time and including the activity that caused the problem is required. So for every entity in SCHEDULED with start times at or after the current time, the GA.USAGE reservation for that activity is erased and the entity destroyed (Figure 3.8). After all are removed, the PERFORM.ACT process relinquishes its CONTACT resource and the PRESCHEDULE routine is called. PRESCHEDULE will start the scheduling process at the CURRENT.TIME, using the activities already performed by the operations simulation process as "history".

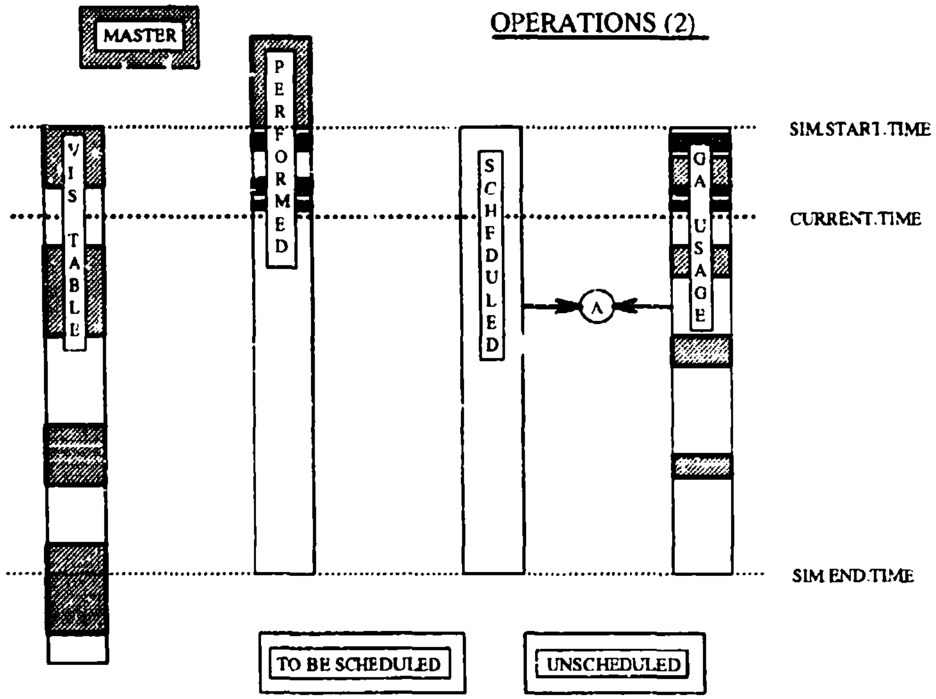

Figure 3.8. Operations Process (2)

3.4.5 Output. Once the scheduling and/or operations processes have completed, the statistics reflecting the performance of the simulation are calculated and saved for analysis. There are five distinct output routines, to allow the simulation operator to tailor the type and amount of output to the current need.

3.4.5.1 REPORT.VIS/REPORT.QUEUES Routines. The first routine dumps the attributes of the VIS.EVNT entities in the VIS.TABLE queue. The rise and set times of the satellite visibility is converted back to days, hours, and seconds from the jminute format of the RISE.TIME and SET.TIME attributes. REPORT.QUEUES performs the same function for the SCHEDULED, TO.BE. SCHEDULED, PERFORMED, and UNSCHEDULED queues. However, only selected attributes of the ACT entities in these queues are reported. In both cases, the data is written to the SIMSCRIPT default output file.

3.4.5.2 REPORT.USE Routine. The REPORT.USE routine outputs the contents of the GA.USAGE array. The result is a table that shows the minute-by-minute use of each of the Ground Antennas for the entire simulation period. This report also goes to the SIMSCRIPT default output file.

3.4.5.3 ANALYSIS Routine. This routine provides the primary data on the results of the simulation run. It makes extensive use of SIMSCRIPT's automatic statistical collection facilities. The output of this routine goes to the file whose name is the contents of the text variable OUT.FILE, which is set during initialization.

First, each minute of the GA.USAGE array for each GA is scanned for the value of the array variable. Separate sets of variables are maintained for individual GAs and for the overall usage. The system automatically calculates statistics for these variables through the use of TALLY statements. A list of simulation parameters are then printed. to permanently label the results to follow. The remarks in the GA outage data file are read and printed to further describe the starting conditions for the simulation. A table is then printed listing the number, maximum, minimum, and standard deviation of four system variables being monitored by TALLY statements. "OFFSET" and "PRIORITY" are collected as the activity entities in the PERFORMED are evaluated for the need for further scheduling in the PRESCHEDULE routine. "GA UTIL" and "GA RESV" are the statistics for the variables UTIL and RESV and were collected earlier in this routine. Histograms for the OFFSET and PRIORITY variables, and then the individual and total GA statistics are reported.

3.4.5.4 VALIDATE Routine. The scheduling process is validated by comparing the scheduling results at a specific time in the past to the actual MCS schedule (produced by scheduling personnel at the 2 SOPS) for that time period. This validation data is the entered from MCS logs into a file called "valact.dat". The format of the file is the same as that for the initial activity data read during initialization, except that the GA at which the contact was historically performed is included with the other activity information. The activities it contains are the first of each type after the time set as the simulation start. The VALIDATE routine first opens the validation data file and the first activity record is read. The activity name and SV from the validation activity are used to search for the first (earliest) similar activity in the SCHEDULED queue. As there is a one-to-one correspondence between the validation activities and the scheduled activities, a matching activity will be found. The absolute value of the difference between the historical and simulation activity start times is recorded. In addition, a counter is incremented when the historical and simulation-scheduled GAs match.

3.5 Validation

The first validation method for the MCS model relies on the expertise and experience of the model-builder. Each of the subfunctions of the model were created to duplicate the corresponding function in the MCS: Initialization, Scheduling, Operations, and Reporting. At each stage in model creation, the builder tested the subfunctions and compared the result to first-hand knowledge of how the like MCS function operates.

The second methodology for the MCS Simulation validation will use detailed MCS data for a period of time prior to and after some fixed date in the past. This date becomes the validation datapoint. The data prior to validation time is required to "hot start" the simulation in any case, but it also allows the simulation to start with the exact state of the MCS at that instant. The

MCS data after the validation time is used to provide the simulation the identical environmental conditions experience! by the MCS, throughout the run of the simulation.

Once the simulation has been run, under conditions matching those experienced by the MCS for the same time period, the simulation schedule is compared to the actual performance of the MCS. Quantitative measures are taken of the differences between the MCS and simulation results. The magnitude of the differences are an indication of the simulation's scheduling "fidelity", that is, its ability to reproduce the MCS generated schedule. An exact reproduction of the MCS schedule would result in zero difference. The measures *do not* indicate the absolute scheduling efficiency or quality of either the MCS personnel or the simulation. The parameters of the simulation can be adjusted to maximize the simulation fidelity (by minimizing the measures), but considering the vast number of possible schedules that meet all the scheduling requirements, a perfect match is not likely.

3.6 Experimentation

Once its validity has been tested, the MCS Simulation will be used to examine the probable response of the MCS to changes in the operating environment. The applicability of the simulation results to the actual system largely depends on the accuracy of the scheduling process. If after the validation step the differences between the simulation and human scheduler are zeroed, then the experimental results could be assumed to be representative of the actual system response under the test conditions. This will not be the case, so the best that can be assumed from the outset is that the test results are consistent among themselves. If this can be demonstrated, then trends observed between experiment results can be assumed to be true also for the MCS, even if the absolute results cannot be compared.

The input parameters used for the experiments are the number of operational satellite vehicles and the number of operational GPS Ground Antennas. The SVs will vary between 16 (the number used for validation) and 24, the maximum number of operational vehicles currently planned. There is no plan to add to the current suite of five GAs, so the number of operational GAs used for testing will vary from three to five. Each combination of three and four operational GAs will be tested. As time allows, alternative scheduling priorities will be tested to determine their affect on simulation results.

As described in Section 3.4.5.3, the simulation results will be a measure of the scheduling effort and the GA utilization. These results will be tested for consistency, correlation, and trends.

IV. Analysis and Results

4.1 Overview.

During the process of coding the simulation model described in Chapter 3, care was taken that the flow of the program emulate the MCS operational process as closely as possible. There were two primary reasons. The first was that this process was well understood, and by mapping the flow of the code to the function of the MCS, it could be clearly seen that the program was executing as expected. Secondly, having the program execution process model the MCS operations process provided reassurance that the simulation would duplicate the results of the MCS with a certain minimal accuracy level. The first reason is related to *verification*, the second to *validation* (11:11). The results of these two critical modeling steps are described in this chapter.

Once the simulation was completed, verified, and validated, a set of experiments were designed to exercise the model in circumstances similar to those encountered by the MCS and provide data for analysis.

4.2 Verification.

In the verification process, the concern is the internal consistency of the model (15:210). This process ensures the model has been constructed properly, according the the rules of the chosen modeling environment. For physical models, this may entail re-measurement of critical dimensions. In mathematical models, perhaps supporting proofs are re-examined for validity. With computer simulation models, the verification process includes a code review and "debug" to ensure the program parameters and logic are as intended.

4.2.1 Verification Analysis. For this computer simulation, verification includes an examination of the SIMSCRIPT II.5 source code to ensure there are no numerical or logical errors. This was performed at the completion of each SIMSCRIPT module, then again just prior to the start of experimentation. As described earlier, this task was simplified by designing the initialization routines to maximize the amount of data stored external to program. For the working data created by the program during it's operation but not routinely output, additional print statements were included in the code during development and validation to make these values external. This allowed not only verification of the numerical correctness of internal variables, but also allowed the program logic to be examined in greater detail and verified. Most of these statements were removed prior to the experiment stage but their vestiges remain in the various REPORT routines retained in the operational code.

As an example of the type of data collected during verification and validation, the results of one early prototype run is shown in Tables 4.1 and 4.2. The first table shows the SIMSCRIPT modules announcing they had been called. When the routine SCHEDULE was manipulating activity entities, the routine printed each action, the current simulation time, and entity attribute data.

Table 4.1. Scheduling Verification/Validation Sample Data.

```
 INIT.ACTS
PRESCHEDULE
SCHEDULE
 PREVIOUSLY: SVN08/Supp.1       ( 15/ 360) scheduled for  323990
  new act: SVN08/Supp.1       ( 15/ 360) scheduled for  324350
   NOW:          SVN08/Supp.1       ( 15/ 360) scheduled for  324350 at KWAJ
 PREVIOUSLY: SVN08/Supp.4       ( 12/ 360) scheduled for  323990
  new act: SVN08/Supp.4       ( 12/ 360) scheduled for  324337
   NOW:          SVN08/Supp.4       ( 12/ 360) scheduled for  324337 at KWAJ
...
PRESCHEDULE
SCHEDULE
 PREVIOUSLY: SVN09/Supp.1       ( 15/ 360) scheduled for  323980
  new act: SVN09/Supp.1       ( 15/ 360) scheduled for  324340
   NOW:          SVN09/Supp.1       ( 15/ 360) scheduled for  324340 at DIEG
   NOW:          SVN10/Supp.1       ( 15/ 360) scheduled for  324330 at ASCN
 PREVIOUSLY: SVN10/Supp.4       ( 12/ 360) scheduled for  323970
  new act: SVN10/Supp.4       ( 12/ 360) scheduled for  324289
```

Table 4.2 shows one way the operations simulation process was verified. As each activity was assigned to a different segment of the operations task flow, output statements in the code printed the time of the activity status change and details about the function and the entity. As seen in

Table 4.2. Operations Verification Sample Data.

```
324000   ASSIGN: Supp.4  , SVN26 at PIKE (5), 324012/12 minutes
  324000   WAIT: Supp.4  , SVN26 at PIKE (5), 324012/12 minutes
324000   ASSIGN: Supp.4  , SVN25 at KWAJ (4), 324013/12 minutes
    324013   PERFORM: Supp.4  , SVN25 at KWAJ (4), 324013/12 minutes
    324014   PERFORM: Supp.3  , SVN20 at CAPE (2), 324014/10 minutes
      324024   DONE: Supp.4  , SVN26 at PIKE (5), 324012/12 minutes
      25   Rescheduling
REPORT USE:   Current time = 324024
```

the last two lines of Table 4.2, the operations routine also notified the operator when it required a schedule change, then printed the status of the system queues.

4.2.2 Verification Results. The model is technically verified, in light of its error-free compilation and execution. Since all the expected scheduling and operational steps occurred as planned, this verifies that the program logic is as designed.

4.3 Validation.

The goal of the validation process is to determine the *fidelity* of the model, how well it reproduces the function of the modeled system in terms of the chosen measurement criteria. The MCS Master Contact Schedule is the master plan for all scheduled MCS activities, so the central theme of the MCS model validation process is how well the model reproduces this key document. In addition to this validation method, the overall performance of the model was evaluated by an MCS-experienced operator. This "face" validation was top-down, as it occurred at each step of the model building process.

4.3.1 Validation Analysis. Two validation principles were applied to test whether the MCS model faithfully reproduced the performance of the actual MCS. The first was applied both during the model-building process and after the model was completed. This was basic *face* validation. This test asks if behavior of the model agrees with that of the real system, in the opinion of knowledgeable experts (4:204). In this case, the system expert is also the model-builder. The expertise was gained through three years of association with the GPS MCS, as both an operational crewmember and subsystem manager. This experience provides the *a priori* knowledge-base from which the model was designed and constructed. This knowledge and experience is also applied to the logical analysis of the model's output. During testing and experimentation of the MCS model, the data provided by the model was continually examined for aptness and reasonability. The model passed these validation tests.

As a second, more quantitative means of testing the validity of the model, a schedule produced by the model for the test date of 10 April 1992 was mathematically compared to the schedule produced and used by the MCS on that date. The hypothesis was that a model that duplicates the MCS scheduling process perfectly would produce the same schedule as the MCS, given identical starting conditions. From the outset it was understood that perfect fidelity is theoretically possible but practically very unlikely. This is due to the very large number of alternative schedules that meet all the criteria of a valid schedule. Often in practice, when faced with equally valid scheduling choices, the human scheduler will apparently decide the result arbitrarily. There may be some personal rule the human uses to choose between two similar choices, but the simulated scheduler has no chance to duplicate either some subtle (and unique) rule or a "mental coin-flip".

Compounding this problem, once a deviation is made in the schedule, the differences tend to cascade with time. For example, the human scheduler at 1000 hours has the equally valid option of using either CAPE or ASCN ground antenna for a 30 minute activity on an SV. She chooses ASCN by some arbitrary means. As the scheduling process continues, an activity on another SV with ASCN visibility is needed at 1020. The earlier choice of ground station now has a bearing on the current decision, as ASCN is no longer available for use. Had the earlier choice been CAPE, the alternative schedule would have ASCN available. Over time, similar small changes eventually result in two quite different schedules, all due to the decision made earlier in the process Slight deviations in the scheduled *time* of activities also has the effect of causing subsequent changes. For these reasons, it is not expected that the MCS model can duplicate the scheduling performance of the MCS schedule makers exactly.

(It is important to note, however, that the model scheduling algorithm has the capability of producing *better* schedules than the human scheduler. A better schedule may make more efficient use of system hardware resources or personnel; may work around system resource outages with fewer "tardy" activities; or may even produce the same quality schedule in less time. Further testing would be required to ascertain the relative effectiveness of automated scheduling methods.)

Given an inherent and unavoidable lack of fidelity, the validation test was constructed to detect the schedule differences over a short time span on the date selected. First, the activities of the prior day (9 April) were collected from the Master Contact Schedule and entered into the initial activity file (described in Section 3.4.2.5). Only the 14 SVs operational on 9/10 April were used. Next, the scheduled GA and MCS outages that occurred on 10 April (also taken from the Master Contact Schedule) were entered into the OUTAGE.DAT file. These resources were "worked around" by the MCS scheduler and would also have to be handled by the model. Finally, the data on first activity of each type for each SV as really scheduled on 10 April was entered into a validation activity file. No further activities would be useful due to the compounding error effect described above.

Under these conditions, the validation run was performed 14 times, each with slight changes to the primary scheduling rule. The modifications changed the order in which the activities to be scheduled were selected for insertion into the tentative schedule. This was done to optimize the performance of the model scheduler in terms of reproducing the MCS schedule. This "tuning" of the model *does not* optimize the scheduler against some absolute criteria; it minimizes the difference between the real system and the model. The validation test inputs and raw results are in Appendix C; these results are summarized in Table 4.3. Overall, the differences between the various ranking methods were negligible, and resulted in the minor reordering of scheduled activities.

Table 4.3. Validation Results

Test	Priority	Max	Min	Mean	Offset	No. of
No.	Scheme	Offset	Offset	Offset	Std Dev	GAs
		(minutes)	(minutes)	(minutes)	(minutes)	Matched
1	hP/hL/hI	148	0	41.1404	40.9769	23
2	hP/hL/lSV	148	0	41.1404	40.9769	23
3	hP/lL/hI	168	0	41.9825	43.0053	25
4	hP/lL/lSV	168	0	41.9825	43.0053	25
5	hP/lSV/lL	148	0	40.8947	41.0565	25
6	hP/lSV/hL	148	0	40.8947	41.0565	25
7	hP/hI	148	0	40.8246	41.0648	25
8	hP/lI	148	0	40.5965	40.8355	26
9	hP/hD	148	0	40.5965	40.8355	26
10	hP/lD	148	0	40.8246	41.0648	25
11	hP/lI/hSV	148	0	40.6667	40.8276	26
12	hP/lI/lSV	148	0	40.5965	40.8355	26
13	hP/lI/hST	148	0	41.1404	40.9769	25
14	hP/lI/lST	168	0	41.4211	42.6972	25

Priority Scheme Key

h - High	l - low
P - PRIORITY	L - LEAD.TIME
I - INTERVAL	SV - SV (name)
ST - START.TIME	D - DURATION

The Priority Scheme column describes the rank ordering of the activity entities in the TO.BE SCHEDULED queue. The scheduling process will try to fit the activities into the schedule in this order. The initial sorting is always by high PRIORITY, to allow the rescheduling priority system to operate. It was anticipated that high LEAD.TIME would be the best starting choice for the second sorting level, as this would allow the activities closest to being tardy to be scheduled first. A third sorting level was included to allow further differentiation of the results. Fourth and greater levels had no effect on the outcome and were removed. The test proceeded by successively changing the second and third ordering criteria in an effort to converge on the simulation result schedule that best matched the MCS schedule. Tests 8 and 9 resulted in the lowest mean differential between the activities in the simulated versus MCS schedule. Additional tests were then performed to test alternative ordering arrangements.

Table 4.4. Activity Types.

SV	Block	Activity Name	Start Time (Jminute)	Duration (minutes)	Interval (minutes)
BI-011	I	SOH	144375	10	480
BII-01	II	GBDDUMP	144220	10	720
BII-01	II	SOH	144210	10	720
BII-12	IIA	NDUTLM	144175	10	720
BII-01	II	NAV	144950	15	1440
BI-011	I	NAV	144385	5	1440
BII-01	II	NAVBUFF	144230	5	1440

The ordering factors that resulted in the best match, low INTERVAL and high DURATION, are not independent. The high duration activities generally have the highest inter-activity interval. The reason the results are identical between Test 8 and 9 are that both orderings minimize the interference between activities being scheduled. Table 4.4 shows the duration/time characteristics for the seven different activity types. Note that the two activities with the lowest duration (Block I NAV and Block II NAVBUFF) have 1440 minute intervals. This means that whether the low INTERVAL or high DURATION ordering mode is selected, the simulation will generally schedule the low INTERVAL activities first. The chronological separation of the successively-scheduled activities reduces contention.

A further indication of this effect is shown by the results of Test 14. When the START.TIME of the previously scheduled activity is used to further sort the activities to be scheduled, the ability of the simulation to match the MCS schedule is reduced. Table 4.5 on Page 4-8 shows this effect in detail, by showing the offsets for each activity in a histogram. The two tests shown, Test 8 and Test 14, are the best and worst, respectively, in terms of mean activity start time differential. As stated above, the differences between the test have only a minor effect on the cumulative statistics. The high PRIORITY/low INTERVAL ordering scheme was used for the experimental runs.

Table 4.5. Validation Test Comparison (Test 8 vs. Test 14).

	Test 8		Test 14	
Start Time Offset (min)	No. of Activities	Percent of Activities	No. of Activities	Percent of Activities
0-10	20	35.09	18	31.58
10-20	7	12.28	9	15.79
20-30	3	5.26	5	8.77
30-40	3	5.26	3	5.26
40-50	4	7.02	3	5.26
50-60	2	3.51	1	1.75
60-70	5	8.77	3	5.26
70-80	0	0	2	3.51
80-90	3	5.26	2	3.51
90-100	4	7.02	4	7.02
100-110	2	3.51	2	3.51
110-120	2	3.51	3	5.26
120-130	0	0	0	0
130-140	0	0	0	0
140-150	2	3.51	1	1.75
150-160	0	0	0	0
160-170	0	0	1	1.75
170+	0	0	0	0

4.3.2 Validation results. The overall assessment of the validation results is that the model ***does not*** reproduce the MCS Master Contact Schedule with sufficient fidelity to use the model for quantitative comparisons.

The model does pass the face validation test, as it does operate in a manner comparable to the MCS operations center. The schedule it produced during validation testing does meet the requirements described in the Navstar Satellite Support Requirements document. However, the quantitative results seen when this schedule is compared numerically to the Master Contact Schedule cannot support the use of the model results for quantitative predictions of MCS behavior. This judgement is based on the fact that

- the average difference between 57 activities was over 40 minutes,
- 18 of 57 activities deviated by more than 60 minutes,

- the validation test looked at only the first scheduled activities, which would yield the *best* results the system could produce,
- the large number of deviations this early in the scheduling process would compound as the process continued.

The opposite side of this coin is that the model *does* perform as the MCS well enough to make general comparisons between the model's and the MCS's response. The successful face validation indicates the pieces of the MCS are all in the model. Even though it cannot be assumed the resulting numbers have meaning, the operation of the model should result in data thet will match and predict trends in MCS resource utilization. On this basis, it was decided to proceed with experimentation. The experimental results would be examined for indications that they conform to the expected performance of the MCS (as well as is known). If sufficient parallels can be found between known or suspected MCS behaviors and model performance, it may be possible to make conclusions about general MCS trends based on the model's output.

4.4 Experimentation.

The MCS model created for this research is not subtle. It is the embodiment of an MCS high-level block diagram. As such, there are not a huge number of experimental parameters that can be modified to perturb the model's operation. The model is also currently dedicated to research involving the two major environmental factors that can and will create problems for the MCS - the growing number of satellites in the GPS constellation and the reliance on a limited network of ground stations to keep those satellites operational. These factors dictate the scope and direction of experimentation.

As described in Section 4.3.2, the output from the model is presumed to be not of sufficient quality to make one-to-one quantitative predictions. The focus of experimentation is to determine if the model can at least be used to make general conclusions about MCS behavior.

4.4.1 Experiment Plan. The major factors that can be varied to exercise the MCS model are:

1. the number of active GPS satellites in the constellation,
2. the number of GPS Ground Antennas available for use, and
3. the number of simultaneous MCS-to-satellite contacts possible.

At the validation date of 10 April 1992, there were 15 fully operational GPS satellites and one operational but undergoing testing (BII-12). The 15 satellites broke down into four aging Block I (R&D) SVs, nine Block II SVs, and two Block IIA SVs. The operational constellation will include 21 operational satellites and three operational spares, 24 total. The MCS is programmed to handle as many 30 satellites simultaneously, in a mixture of Block I, II, and IIA vehicles. For the purposes of this experiment, the number of satellites was varied between the current number (16) and the operational number (24), keeping the four Block I's and adding eight Block IIA's. If the four currently operational Block I satellites remain healthy and the Air Force Space Command's goal of six launches per year is met, this 24 satellite configuration will occur sometime in 1993.

There are currently no plans to augment the number of GPS Ground Antennas now operational. However, it often happens that one or more GAs are unavailable for satellite contacts for extended periods. In the event of a natural disaster or catastrophic failure, any GA could be out of service for days or even weeks. Another circumstance is that the CAPE GA is reserved for prelaunch compatibility test for days prior to a GPS satellite launch. So the number of operational GAs is an experiment parameter. Each GA was removed from the system in turn, then removed in pairs. The exception to this rule is the PIKE GA. It is a unique case in both the operation of the model and in the experiment matrix. As a non-dedicated GPS resource, it is normally used far less than the remaining GAs. This is accounted for in the model by making PIKE last on the list for selection when the scheduler looks for open GAs. PIKE also covers nearly the same coverage area as CAPE, as they are only 10 degrees apart in latitude and 25 degrees separated in longitude.

Therefore, assuming PIKE outages have minimal effect on MCS operations, it was excluded from paired outage table. This reduced the number of this type of experiment from ten to six.

While the MCS is designed to handle up four satellite contacts at one time, the current operational practice is to have a maximum of only three contacts. This is due to crew manning (each crew has three Satellite Support Operator personnel trained and available to handle routine contacts) and MCS computer system loading. Due to the large number of experiments planned with just the number of SVs and GAs being varied, it was decided to fix the maximum number of contacts to three and not vary this parameter experimentally.

More significantly, the stochastic nature of the operations segment of the simulation was not included in this set of experiments. This is due in part to insufficient time to perform experiments with this facet of the simulation. However, the reason these tests were of lesser importance was that the data collected at the MCS and discussion with the MCS crew indicated that variable activity durations were not a significant MCS operational factor. The standard activity durations listed in the NSSR were so conservative that seldom if ever did the activity durations as performed exceed the scheduled times allowed. In the rare instances where this occurred, the variation was due to system anomalies that disrupted more than just the activity being performed. The wide variety of anomalies that create this situation and their diverse effect on operations are not practical to include in this simple model.

The experimental suite required results from 12 different GA outage configurations and 9 SV quantities, for total of 108 distinct runs of the MCS model. To keep the simulation run time reasonable, the simulation length was limited to 48 hours. The initialization configuration used during validation was the experimental baseline. This configuration corresponds to the state of the OCS as of 0000Z 10 April 1992. As each SV was added to the configuration, its activities on 9 April were included in the activity initialization file, as was it's GA visibility data. In the case of SVs not yet launched, an fictional activity history consistent with the "real" activities was created. The input conditions during testing are described more fully and the data itself provided in Appendix B. The results of the experiments are summarized in the next section, in both tabular and graphic formats.

4.4.2 Experiment Analysis. The data provided by the experiment runs fall into two general categories, based on the measurement criteria selected to characterize the MCS model performance. The first category is data that describes the activity scheduling process and the number of missed/failed satellite support requirements. The second category of data relates to the utilization of OCS resources, primarily the GPS Ground Antennas.

4.4.2.1 Activity Scheduling Analysis. The most significant indicator of the effect of constellation size and GA outages on scheduling ability is the number of activities that could not be scheduled. This number, along with the total number of activities scheduled for that experiment, is shown af'er the slash in Table 4.6. No slash indicates there were no activities that could not be schedul.u.

Table 4.6. Number of Activities Scheduled/Failed

	Ground Anten .a(s) Unavailable											
SVs	NONE	ASCN	CAPE	DIEG	KWAJ	FAKE	A/K	A/C	A/D	K/D	C/D	C/K
16	184	184	184	184	184	184	184	185	187/1	181/3	184	184
17	196	196	196	196	196	196	196	198	199/1	193/3	196	196
18	208	208	208	208	208	208	208	210	210/1	205/3	208	208
19	220	220	220	221	220	220	220	222	226/1	216/3	222	220
20	232	232	232	233	232	232	232	234	238/1	231/3	234	232
21	244	245	244	245	246	244	246	247	250/3	245/3	246	246
22	256	257	256	257	258	256	258	259	262/3	257/3	258	259
23	268	269	268	269	270	268	270	271	274/3	269/3	270	271
24	280	281	280	281	283	280	284	283	286/4	284/3	282	284

As seen in the NONE column, under optimal conditions there were 12 additional satellite activities added to the schedule with each additional satellite added to the constellation. As GA resources were removed from availability, the number of activities paradoxically increased, especially when the system had to cope with a large constellation. Follow along the bottom row of Table 4.6 to see this effect. The reason this occurs is the scheduling algorithm. The system schedules by looking first for the latest time an activity can be performed. If that time is unavailable, the system "slides" the activity along the schedule looking for the first earlier time resources are available for the activity.

For example, an activity is normally scheduled four hours apart. The system determines the last time this activity was scheduled and starts looking four hours hence to schedule the next occurrence. If that time is unavailable, it may have to settle for a time only *three* hours after the last occurrence. The repetition rate of this activity has thus increased. because the effective time interval between activities of this type *as scheduled* has decreased. If this happens four times over the 48 hour simulation period, the last scheduled activity of this type will occur four hours earlier than in a schedule without conflict. Thus there will be "room" for the scheduler to add another activity of this type to the schedule, increasing the total activity count. This effect can be put to use as an indicator of scheduling "stress", or how much conflict occurs in the scheduling process.

Due to the periodic nature of the scheduled activities, the scheduler will not experience continuous conflict. The various activities will occasionally synchronize and for a short period the system will have trouble finding resources to satisfy the scheduling requirements. This situation is shown clearly in Figure 4.1 on Page 4-14, which displays the number of GAs utilized over simulation time. The periodic nature of scheduling "crunches" is clear, with the most serious problems coming at about 1400 and 2800 minutes into the simulation. Note that increasing the constellation size from 16 to 24 satellites makes the problem worse; most of the additional activities (dotted lines) are clustered around previous clusters.

Using the number of activities scheduled as an indicator, note that both CAPE and PIKE outages cause no increase in stress. This is not unexpected, as these GAs cover approximately the same segment of the satellite orbits and can "cover" for each other (as seen in Figure 2.4). The system first experiences noticeable stress when the Diego Garcia GA is out and there are 21 satellites in the constellation. This is expected, also, as DIEG has the least overlap of all the GAs.

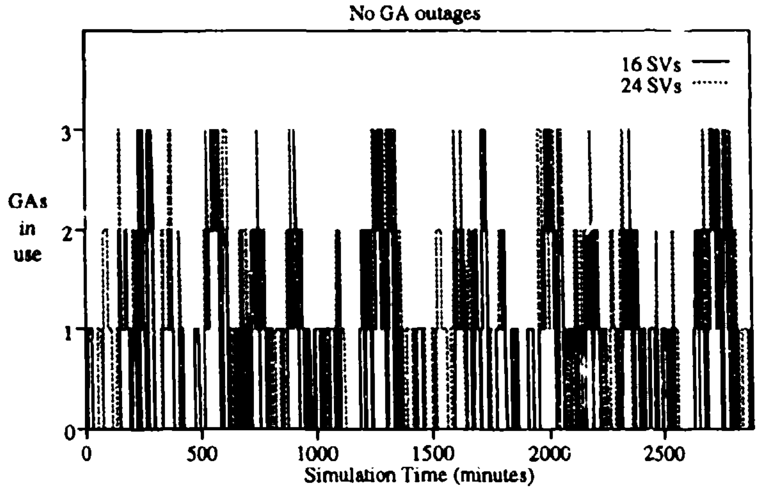

Figure 4.1. Number of GAs In Use vs. Simulation Time.

The system has fewer options with DIEG down. The fact is reinforced by examining the dual outage scenarios in Table 4.6. The only scenarios with activities failing to be scheduled involve DIEG and either of the adjacent GAs. (The number of activities scheduled in these columns naturally does not include the failed supports.) In these dual-outage scenarios, scheduling conflicts are greater and begin earlier than with the single outage scenarios.

Tl e next three tables describe other indicators of scheduling conflicts. Table 4.7 shows the summed priority of all the scheduled activities. As described in Section 3.4.3.3 (Page 3-19), priority is an activity entity attribute used to improve the activity's probability of being scheduled in a conflict situation. The higher the total priority, the more conflicts are occurring. These values give a better indication of the level of scheduling difficulty for each scenario than just the activities scheduled. Of interest in the table is the fact that every run had to use the priority system at least once. A review of the raw simulation results showed that this occurred only 20 minutes into the scheduling period in the first scheduling iteration. Two BII-12 activities were trying to be scheduled for this time while this SV had visibility only at DIEG. The activity with the low interval (SOH) had default priority, but the other activity (NDUTLM) had to start between 0000 and 0020 hours or exceed its interval requirement. The first iteration bumped the NDUTLM activity ahead of the SOH and both were scheduled during the next scheduling attempt.

Table 4.7. Total Activity Priority.

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	1	1	1	9	1	1	1	14	19	13	9	1
17	1	1	1	9	1	1	1	14	19	13	9	1
18	1	1	1	9	1	1	1	14	19	19	9	1
19	1	1	1	9	1	1	1	14	16	19	9	1
20	1	1	1	9	1	1	1	14	16	19	9	1
21	1	11	1	9	1	1	11	32	25	28	9	1
22	1	11	1	9	1	1	11	32	25	28	9	1
23	1	11	1	9	1	1	11	32	25	28	9	1
24	1	11	1	9	3	1	16	32	25	33	9	3

Another interesting window into the inner workings of the scheduling algorithm is the decrease in total priority in the ASCN/DIEG outage scenario as constellation size increases from 18 to 19. Note on Table 4.6 that at this point there is an increase of 16 activities scheduled, the largest change of any for this statistic. What has happened is that a time shift occurred for one type of activity early in the schedule-building process that has allowed subsequent activities to be scheduled with less difficulty than before. The small increase in total interval offset (see below) may indicate a scheduling tradeoff has occurred; one type of activity being pushed very tardy while others are allowed to return closer to their latest "legal" start time.

As seen first in Table 4.6, Table 4.7 shows that the scheduler has the hardest time when DIEG is out, alone or with another GA. When two adjacent GAs are out, the difficulty peaks. While PIKE can pick up the contact load when CAPE is out along with DIEG or KWAJ, PIKE is not much help when ASCN and CAPE are out together. This is because ASCN and CAPE are adjacent and their outage leaves too big a hole in the coverage. Note that the priority level of a CAPE/DIEG or CAPE/KWAJ outage stays fairly level. PIKE utilization should be increasing rapidly during this scenario as the constellation size grows. The dual coverage effect between PIKE and CAPE is also seen in that the CAPE/DIEG and CAPE/KWAJ total priorities are identical to DIEG and KWAJ alone.

Table 4.8 shows the response of the next level of activity conflict, the interval offset. This attribute is used to extend the scheduling window into the "tardy" time period, in an effort to schedule the activity after the priority system has failed. The values indicate the total number of minutes the system had to add to scheduled activities to complete the schedule. Failed activities are not counted. Table 4.9 shows the maximum interval offset the system had to apply. The synergistic effect of the ASCN/DIEG and KWAJ/DIEG outages on scheduling difficulty are also seen in these tables.

Table 4.8. Total Activity Interval Offset

	Ground Antenna(s) Unavailable											
SVs	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	0	0	0	420	0	0	0	130	498	980	420	0
17	0	0	0	420	0	0	0	130	530	980	420	0
18	0	0	0	420	0	0	0	130	523	1120	420	0
19	0	0	0	420	0	0	0	130	540	1180	420	0
20	0	0	0	420	0	0	0	130	540	1180	420	0
21	0	140	0	420	0	0	140	270	930	1390	420	0
22	0	140	0	420	0	0	140	270	930	1390	420	0
23	0	140	0	420	0	0	140	270	930	1390	420	0
24	0	140	0	420	0	0	140	270	930	1410	420	0

Table 4.9. Maximum Activity Interval Offset

	Ground Antenna(s) Unavailable											
SVs	System	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	0	0	0	200	0	0	0	40	210	510	200	0
17	0	0	0	200	0	0	0	40	210	510	200	0
18	0	0	0	200	0	0	0	40	210	510	200	0
19	0	0	0	200	0	0	0	40	210	510	200	0
20	0	0	0	200	0	0	0	40	210	510	200	0
21	0	70	0	200	0	0	70	70	390	510	200	0
22	0	70	0	200	0	0	70	70	390	510	200	0
23	0	70	0	200	0	0	70	70	390	510	200	0
24	0	70	0	200	0	0	70	70	390	510	200	0

4.4.2.2 Activity Scheduling Results. The above data confirms the basic assertions made earlier regarding the criticality of the Diego Garcia GA. Using these results, system managers could rank the GPS GAs by least desirable to be unavailable. The order would be (in order of most preferable to be unavailable)

1. CAPE or PIKE (tie)
2. KWAJ or CAPE/KWAJ (tie)
3. ASCN
4. ASCN/KWAJ
5. DIEG or CAPE/DIEG
6. ASCN/CAPE
7. ASCN/DIEG
8. KWAJ/DIEG

From a practical standpoint it is never preferable to have two GAs down at the same time, but the simulation indicates it will be easier to build a schedule (and of course perform one) if both ASCN/KWAJ, ASCN/CAPE, or CAPE/KWAJ were down than if DIEG alone were unavailable. This is with PIKE fully available but used only if no GPS-dedicated resources are available.

These data build a case for the increased confidence in the validity of the model's operation. The satellite contact scheduling process appears to react in tune with the constraints placed on the system by the geographic location of the ground stations. In addition, the scheduling algorithm reacts logically to the increased stress placed on the system by the GA outages and constellation build-up.

4.4.2.3 Ground Antenna Utilization Analysis. The overall system usage of the GAs for each experimental scenario is summarized in Table 4.10; the standard deviations of total utilization are contained in Table 4.11. The means represent the mean number of GAs in use for any given minute during the simulation, and vary from a low of 0.7000 to a high of 1.1302. These numbers do not count the GAs unavailable during that experiment. The means relate directly to the number of activities scheduled during that run, because there are a fixed number of GA-minutes available during the simulation run and adding activities to the schedule use these up. This is the reason for the regular increase in mean utilization as the simulated constellation grows. The standard deviations are simply a confidence builder; their uniformity is reassurance the model is not off the rails in some fashion that is hidden by the other indicators.

Table 4.10. Mean Number of GAs Utilized for SV Activities (per simulation minute).

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	0.7132	0.7132	0.7132	0.7132	0.7132	0.7132	0.7132	0.7170	0.7247	0.7000	0.7132	0.7132
17	0.7625	0.7625	0.7625	0.7625	0.7625	0.7625	0.7625	0.7701	0.7740	0.7493	0.7625	0.7625
18	0.8118	0.8118	0.8118	0.8118	0.8118	0.8118	0.8118	0.8194	0.8309	0.7986	0.8118	0.8118
19	0.8611	0.8611	0.8611	0.8649	0.8611	0.8611	0.8611	0.8688	0.8840	0.8556	0.8686	0.8611
20	0.9104	0.9104	0.9104	0.9142	0.9104	0.9104	0.9104	0.9181	0.9333	0.9049	0.9181	0.9104
21	0.9597	0.9635	0.9597	0.9635	0.9674	0.9597	0.9674	0.9712	0.9826	0.9618	0.9674	0.9674
22	1.0090	1.0128	1.0090	1.0128	1.0167	1.0090	1.0167	1.0205	1.0319	1.0111	1.0167	1.0205
23	1.0583	1.0622	1.0583	1.0622	1.0660	1.0583	1.0660	1.0698	1.0812	1.0604	1.0660	1.0698
24	1.1076	1.1115	1.1076	1.1115	1.1191	1.1076	1.1229	1.1191	1.1302	1.1212	1.1153	1.1229

Table 4.11. Standard Deviation of Number of GAs Utilized for SV Activities.

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	0.8431	0.8039	0.7904	0.7640	0.8039	0.7877	0.7829	0.7150	0.7380	0.7685	0.7442	0.7484
17	0.8473	0.8083	0.7962	0.7691	0.8130	0.7905	0.7922	0.7303	0.7665	0.7797	0.7494	0.7378
18	0.8412	0.8041	0.7897	0.7660	0.7997	0.7840	0.7836	0.7355	0.7854	0.7913	0.7565	0.7265
19	0.8401	0.8376	0.7868	0.7743	0.7985	0.7827	0.8003	0.7556	0.8146	0.8270	0.7946	0.7223
20	0.8590	0.8730	0.8171	0.7812	0.8377	0.8141	0.8285	0.7942	0.8386	0.8660	0.8102	0.7442
21	0.8444	0.8790	0.8018	0.7899	0.8167	0.7987	0.7978	0.8055	0.8484	0.8539	0.8184	0.7357
22	0.8831	0.8840	0.8469	0.8276	0.8277	0.8469	0.8056	0.8241	0.8868	0.8591	0.8332	0.7425
23	0.9231	0.8990	0.8886	0.8428	0.8611	0.8812	0.8361	0.8319	0.8762	0.8830	0.8386	0.7305
24	0.9285	0.9074	0.8899	0.8743	0.8917	0.9054	0.8512	0.8572	0.9102	0.9038	0.8602	0.7520

To associate reality to experimental mean utilizations, a similar calculation was made from the actual Master Contact Schedules for 15, 16, and 17 April 1992. These days were chosen for the lack of apparent OCS anomalies during this period. The GA contact durations for the satellites and satellite activities were totaled, with some approximations made to account for the presence of non-simulated activities mixed-up in the total contact duration. With 141 total contacts counted (each with one, two, or three activities), the total GA use came to 3470 GA-minutes. Since for the three-day period analyzed there were 1440 minutes/day * 3 days * 5 GAs, there were 21600 GA-minutes available for use. Thus the calculated utilization rate for this three day period is 0.1606. For the 16-satellite simulation with no outages, there was a mean of 0.7132 GAs used per simulation minute. But with five GAs operational, each simulation minute = five GA-minutes, so the simulation GA utilization rate is 0.7132 per 5 GA-minutes, or 0.1426. These numbers are meant for general comparison only, as neither calculation method is particularly rigorous, but the "ballpark" figure indicates the simulation is operating as it was designed.

Page 4-20 through Page 4-31 show the response of individual GAs to the changes in both GA outages and constellation size. A graph is shown with each table to assist the reader in discerning trends and unusual behaviors. Notes associated with each table and graph highlight the salient points of that set and inter-relationships between the data shown and other experimental results.

Table 4.12. Individual GA Utilization with No GA Outages.

	Percentage of total simulation time					
	No GAs	ASCN	CAPE	DIEG	KWAJ	PIKE
SVs	in use	in use	in use	in use	in use	in use
16	50.59	16.22	27.74	13.51	11.01	2.85
17	47.19	18.16	28.47	13.51	12.71	3.40
18	42.18	20.10	28.47	13.51	15.69	3.40
19	39.24	21.63	28.68	16.70	15.69	3.40
20	36.53	23.16	28.68	17.12	18.68	3.40
21	32.50	26.13	28.68	17.12	20.63	3.40
22	31.56	27.99	29.06	18.26	22.19	3.40
23	31.35	27.99	30.52	20.21	22.95	4.17
24	29.62	28.99	31.46	20.21	25.94	4.17

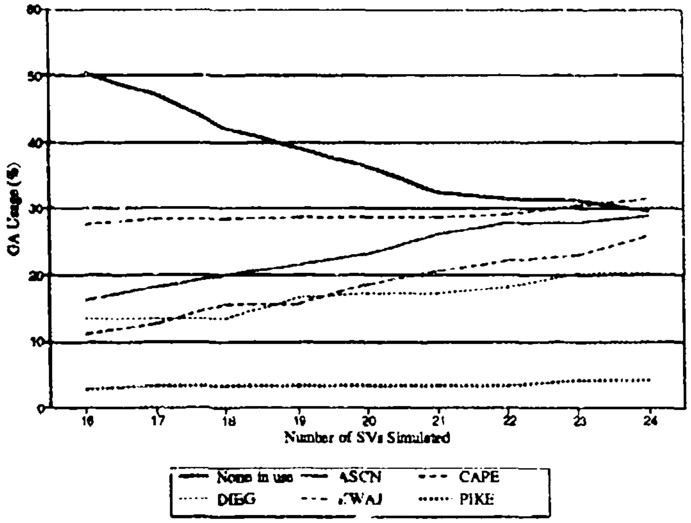

Figure 4.2. GA Utilization - No GA out.

This is the individual GA utilization rates with no outages. The heavy black line shows the percentage of total simulation time there were no GAs in use. This gives a rough idea of the total GA slack in the system. CAPE GAs consistently high utilization is due to the low use of PIKE; activities in the coverage area of both GAs will go to CAPE first. PIKE is last in the GA selection priority to simulate its AFSCN schedule requirements. The utilization of the remaining GAs increases steadily as the constellation grows. It is possible that the satellite activities naturally "cluster" (see Figure 4.1) while in CAPE's coverage area. This would explain CAPE's consistently high utilization.

Table 4.13. Individual GA Utilization with ASCN Out.

	Percentage of total simulation time						Change from "No Outage"				
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	CAPE Δ%	DIEG Δ%	KWAJ Δ%	PIKE Δ%
16	48.47	0.00	31.60	22.22	12.74	4 76	-2 12	3.86	8.71	1.73	1.91
17	45.07	0.00	32.33	24.17	14.44	5.31	-2.12	3.86	10.66	1.73	1.91
18	40.73	0.00	32.33	26.11	17.43	5.31	-1.45	3.86	12.60	1.74	1.91
19	39.27	0.00	34.27	29.10	17.43	5 31	0.03	5.59	12.40	1.74	1.91
20	37.50	0.00	34.65	29.10	20.45	6.84	0.97	5.97	11 98	1.77	3.44
21	35.03	0.00	35.42	31.70	22.40	6.84	2.53	6.74	14.58	1.77	3.44
22	32.81	0.00	37.26	32.85	23.96	7.22	1.25	8.20	14.59	1.77	3.82
23	31.28	0.00	38.72	34.79	25.10	7.60	-0.07	8.20	14.58	2.15	3.43
24	28.99	0.00	39.10	36.74	28.09	7.22	-0.63	7.64	16.53	2.15	3.05

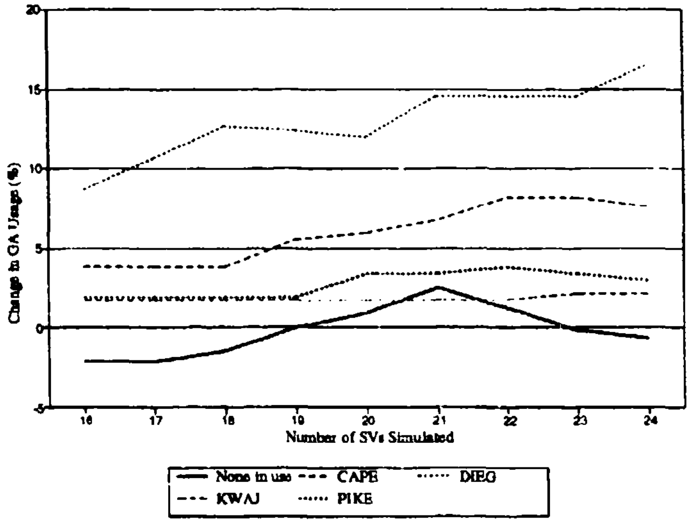

Figure 4.3. GA Utilization Change - ASCN out.

With ASCN GA unavailable, the adjacent GAs (DIEG to the east and CAPE to the west) pick up the slack. CAPE had less slack to start, so most of the additional load falls to DIEG. The extensive overlap of CAPE and DIEG into ASCN coverage means the slack is picked up "clean", with little additional load slipping to KWAJ and PIKE. The "None in Use" line indicates that with 20, 21, and 22 SVs simulated, the system is actually *more* efficient than the baseline in terms of more time with no GAs in use.

Table 4.14. Individual GA 'Jtilization with CAPE Out.

	Percentage of total simulation time						Change from "No Outage"				
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	ASCN Δ%	DIEG Δ%	KWAJ Δ%	PIKE Δ%
16	46.98	16.98	0.00	14.27	17.90	20 17	-3 61	0.76	0 76	8 69	17 32
17	43.65	18 92	0.00	14.27	21.94	21.11	-3.54	0.76	0.76	9.23	17 71
18	39.62	20.67	0.00	14.27	24.93	21.11	-2.56	0.77	0.76	9.24	17.71
19	35.82	22.60	0.00	17.47	24 93	21 11	-3 42	0.97	0.77	9.24	17 71
20	34.72	24 13	0.00	17.88	27.92	21.11	-1 81	0 97	0.76	9.24	17.71
21	30.69	27.12	0 00	17.88	29.86	21 11	-1.81	0 97	0 76	9.23	17.71
22	30.07	29.34	0.00	18.65	32.19	20.73	-1.49	1 35	0.39	10 00	17.33
23	29.86	29.34	0.00	20.59	34.20	21.70	-1.49	1.35	0.38	11.25	17.53
24	27.78	30.90	0.00	20.42	37.19	22.26	-1 84	1 91	0.21	11.25	18.09

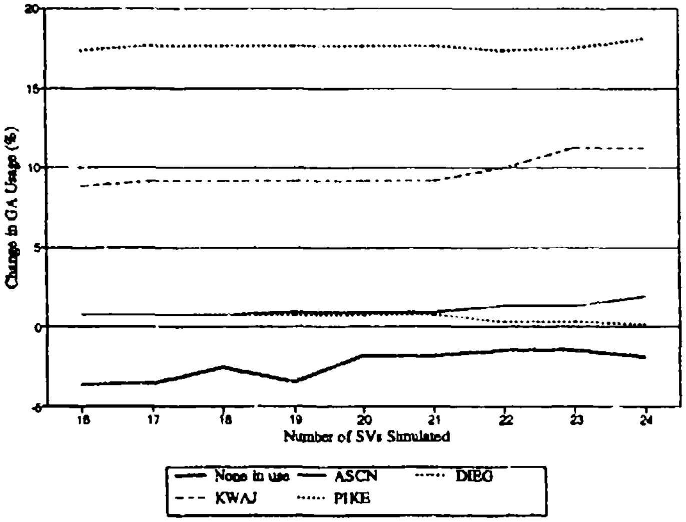

Figure 4.4. GA Utilization Change - CAPE out.

In this scenario, PIKE picks up the majority of the load created by a CAPE outage. CAPE and PIKE coverage overlaps to such an extent and CAPE leaves such a large hole that the system overcomes its inherent reluctance to schedule activities at PIKE. As they are to the east, ASCN and DIEG are on the "back" of PIKE/CAPE coverage and the loading problem is solved before their visibility periods start for the west-to-east travelling satellites.

Table 4.15. Individual GA Utilization with DIEG Out.

	Percentage of total simulation time						Change from "No Outage"				
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	ASCN Δ%	CAPE Δ%	KWAJ Δ%	PIKE Δ%
16	46.11	25.24	29.27	0.00	12.43	4.38	-4.48	9.02	1.53	1.42	1.53
17	42.74	27.19	30.00	0.00	14.13	4.93	-4.45	9.03	1.53	1.42	1.53
18	38.72	29.13	30.21	0.00	17.12	4.72	-3.46	9.03	1.74	1.43	1.32
19	35.56	31.25	31.18	0.00	19.34	4.72	-3.68	9.62	2.50	3.65	1.32
20	32.50	32.78	31.18	0.00	22.74	4.72	-4.03	9.62	2.50	4.06	1.32
21	30.35	35.76	31.18	0.00	24.69	4.72	-4.03	9.62	2.50	4.06	1.32
22	28.75	38.37	31.56	0.00	26.63	4.72	-2.81	10.38	2.50	4.44	1.32
23	26.74	39.51	33.78	0.00	27.01	5.90	-4.61	11.52	3.26	4.06	1.73
24	26.70	40.52	34.34	0.00	30.38	5.90	-2.92	11.53	2.88	4.44	1.73

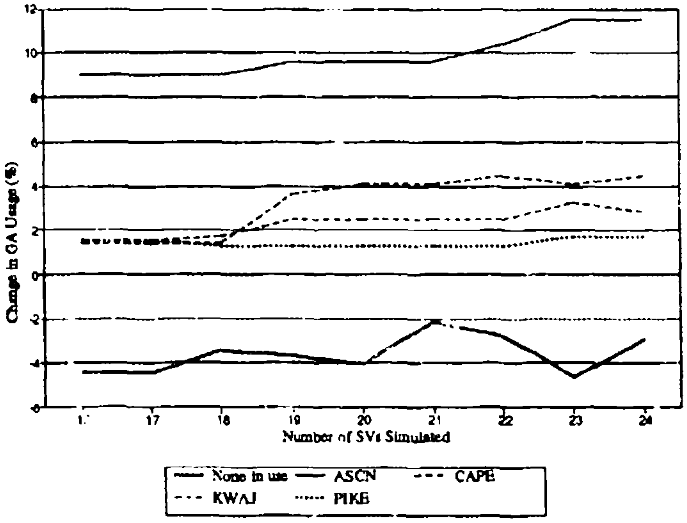

Figure 4.5. GA Utilization - DIEG out.

With DIEG out, the GA that overlaps its coverage to the west, ASCN, picks up the slack. The response is similar to the ASCN-out scenario, with ASCN and DIEG taking each others' role.

Table 4.16. Individual GA Utilization with KWAJ Out.

	Percentage of total simulation time						Change from "No Outage"				
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	ASCN Δ%	CAPE Δ%	DIEG Δ%	PIKE Δ%
16	48.33	15.28	29.62	17.33	0.00	9.10	-2.26	-0.94	1.88	3.82	6.25
17	45.31	17.22	30.35	17.33	0.00	11.35	-1.88	-0.94	1.88	3.82	7.95
18	40.73	19.17	30.35	17.33	0.00	14.34	-1.45	-0.93	1.88	3.82	10.94
19	37.15	20.69	30.56	20.52	0.00	14.34	-2.09	-0.94	1.88	3.82	10.94
20	35.97	22.01	31.88	20.38	0.00	16.77	-0.56	-1.15	3.20	3.26	13.37
21	31.04	24.24	33.40	22.33	0.00	16.77	-1.46	-1.91	4.72	5.21	13.37
22	28.85	27.22	33.40	22.33	0.00	18.72	-2.71	-0.77	4.34	4.07	15.32
23	27.88	26.88	33.75	24.83	0.00	21.15	-3.47	-1.11	3.23	4.62	16.98
24	27.29	27.85	33.96	24.83	0.00	25.28	-2.33	-1.14	2.50	4.62	21.11

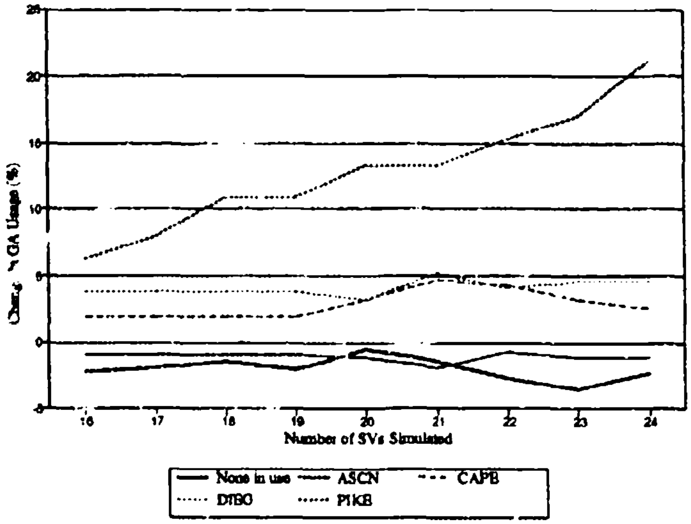

Figure 4.6. GA Utilization - KWAJ out.

Although DIEG picks up some additional use, PIKE's low utilization acts like a magnet to gather the activities normally scheduled for KWA. ASCN use decreases slightly as some of the previous KWAJ (now PIKE) activities are satisfied earlier and skip over ASCN.

Table 4.17. Individual GA Utilization with PIKE Out.

	Percentage of total simulation time						Change from "No Outage"				
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	ASCN Δ%	CAPE Δ%	DIEG Δ%	KWAJ Δ%
16	47.60	16.98	30.89	14.27	9.48	0.00	-2.99	0.76	2.86	0.76	-1.535
17	44.06	18.92	32.12	14.27	10.94	0.00	-3.13	0.76	3.65	0.76	-1.77
18	40.03	20.87	32.12	14.27	13.92	0.00	-2.15	0.77	3.65	0.76	-1.77
19	36.46	22.40	32.33	17.47	13.92	0.00	-2.78	0.77	3.66	0.77	-1.77
20	36.35	23.92	32.33	17.88	16.91	0.00	-1.18	0.76	3.65	0.76	-1.77
21	31.32	26.91	32.33	17.88	18.85	0.00	-1.18	0.76	3.65	0.76	-1.78
22	30.94	28.76	32.71	18.66	20.80	0.00	-0.62	0.76	3.65	0.39	-1.39
23	30.03	29.31	34.17	20.59	21.77	0.00	-1.32	1.32	3.65	0.38	-1.18
24	29.17	30.31	35.10	20.59	24.76	0.00	-0.45	1.32	3.64	0.38	-1.18

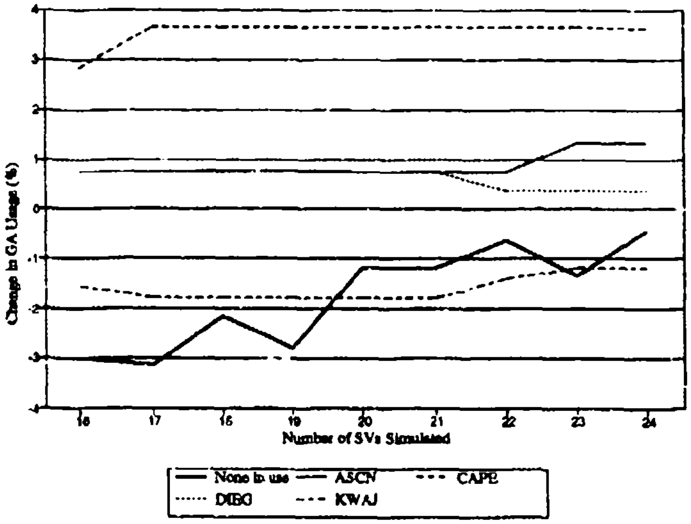

Figure 4.7. GA Utilization - PIKE out.

No GA has to work much harder to compensate for a PIKE outage. The system doesn't seriously require this GA with CAPE available (a crucial "if").

Table 4.18. Individual GA Utilization with ASCN and CAPE Out.

	Percentage of total simulation time						Change from "No Outage"			
	No GAs	ASCN	CAPE	DIEG	KWAJ	PIKE	No GAs	DIEG	KWAJ	PIKE
SVs	in use	in use	in use	in us	in use	in use	Δ%	Δ%	Δ%	Δ%
16	42.60	0.00	0.00	24 17	20.87	26.67	-7.99	10.66	9.86	23.82
17	39.66	0.00	0.00	26.49	23.30	27.22	-7.54	12.98	10.59	23.82
18	36.56	0.00	0.00	28.44	26.28	27.22	-5.62	14.93	10.59	23.82
19	34.56	0.00	0.00	31 42	26.28	29.17	-4.69	14.72	10.59	25.77
20	33.82	0.00	0.00	31.25	28.92	31.63	-2 71	14.13	10.24	28.23
21	30.63	0.00	0.00	33.47	31.42	32.22	-1.87	16.35	10.79	28.82
22	29.31	0.00	0.00	34.24	33.37	34.44	-2.25	15.98	11.18	31.04
23	27.71	0.00	0.00	36.18	36.35	34.44	-3.64	15.97	13.40	30 27
24	26.49	0.00	0.00	37.74	39.31	34.86	-3.13	17.53	13.37	30.69

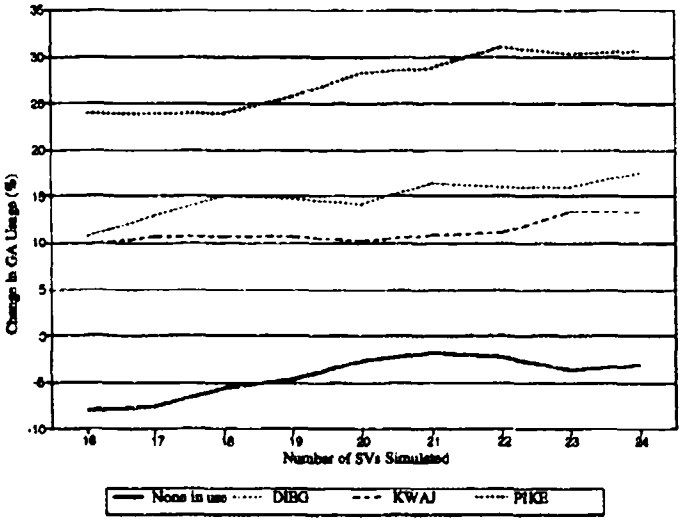

Figure 4.8. GA Utilization - ASCN and CAPE out.

With both ASCN and CAPE out, PIKE is brought to high utilization, together with major increases in the remaining GAs.

Table 4.19. Individual GA Utilization with ASCN and DIEG Out.

	Percentage of total simulation time						Change from "No Outage"			
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	CAPE Δ%	KWAJ Δ%	PIKE Δ%
16	43.51	0.00	40.59	0.00	20.28	11.60	-7.08	12.65	9.27	8.75
17	41.81	0.00	43.26	0.00	21.98	12.15	-5.38	14.79	9.27	8.75
18	38.51	0.00	44.83	0.00	25.35	12.92	-3.67	16.36	9.66	9.52
19	36.77	0.00	45.38	0.00	27.19	15.83	-2.47	16.70	11.50	12.43
20	35.03	0.00	46.91	0.00	30.59	15.83	-1.50	18.23	11.91	12.43
21	33.13	0.00	47.67	0.00	34.38	16.22	0.63	18.99	13.75	12.82
22	32.01	0.00	49.90	0.00	35.94	17.36	0.45	20.84	13.75	13.96
23	28.85	0.00	52.71	0.00	35.94	19.48	-2.50	22.19	12.99	15.31
24	28.09	0.00	52.01	0.00	39.83	21.18	-1.53	20.55	13.89	17.01

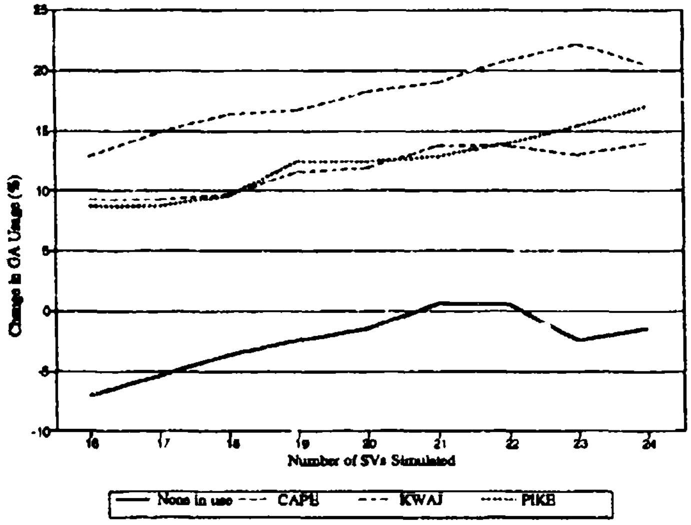

Figure 4.9. GA Utilization Change - ASCN and DIEG out.

As Table 4.6 shows, with this pair unavailable the system experiences major scheduling problems. CAPE exceeds 50% utilization trying to compensate for the outages. This is the first scenario where any GA exceeds even 40%. PIKE coverage overlaps ASCN slightly and DIEG not at all, so its low utilization cannot save the day.

Table 4.20. Individual GA Utilization with ASCN and KWAJ Out.

	Percentage of total simulation time						Change from "No Outage"			
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	CAPE Δ%	DIEG Δ%	PIKE Δ%
16	40.35	0.00	34.24	25.49	0.00	11.60	-4.24	6.50	11 98	8.75
17	·3.33	0.00	34.97	27.43	0.00	13.85	-3.86	6.50	13.92	10.45
18	38.92	0.00	34.97	29.37	0.00	16.84	-3.26	6.50	15.86	13.44
19	36.81	0.00	36.53	32.36	0.00	17.22	-2.43	7.85	15 66	13.82
20	35.49	0.00	36.32	32.78	0.00	21.94	-1.04	7.64	15.66	18.54
21	29.79	0.00	37.85	36.94	0.00	21.94	-2.71	9.17	19.82	18.54
22	27.43	0.00	40.07	36.94	0.00	24.65	-4.13	11.01	18.68	21.25
23	26.74	0.00	41.53	38.69	0.00	26.18	-4.61	11.01	18.68	22.01
24	24.90	0.00	42.29	39.69	0.00	30.31	-4.72	10.83	19.48	26.14

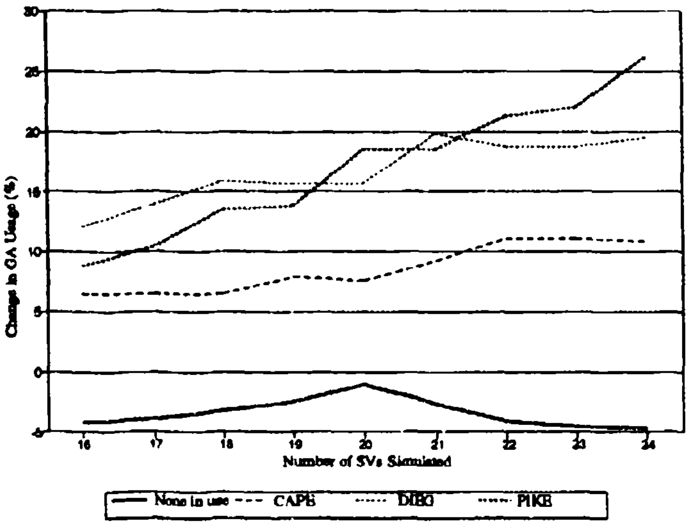

Figure 4.10. GA Utilization Change - ASCN and KWAJ out.

PIKE does better at covering a KWAJ outage than a DIEG outage, so the system can this scenario with more grace. CAPE shoulders ASCN's load and DIEG fills in the gaps on both sides.

Table 4.21. Individual GA Utilization with CAPE and DIEG Out.

	Percentage of total simulation time						Change from "No Outage"			
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in usc	PIKE in use	No GAs Δ%	ASCN Δ%	KWAJ Δ%	PIKE Δ%
16	44.79	26.98	0.00	0.00	27.40	16.94	-5.80	10.76	16.39	14.09
17	41.42	28.92	0.00	0.00	29.06	18.26	-5.77	10.76	16.35	14.86
18	37.78	30.87	0.00	0.00	31.49	18.82	-4.40	10.77	15.80	15.42
19	35.59	35.03	0.00	0.00	33.02	18.82	-3.65	13.40	17.33	15.42
20	33.62	36.56	0.00	0.00	36.42	18.82	-2.71	13.40	17.74	15.42
21	31.67	41.56	0.00	0.00	38.37	18.82	-0.83	15.41	17.74	15.42
22	29.51	42.53	0.00	0.00	40.31	18.82	-2.05	14.54	18.12	15.42
23	26.81	44.27	0.00	0.00	41.39	20.94	-4.54	16.28	18.44	16.77
24	25.42	45.66	0.00	0.00	44.55	21.32	-4.20	16.67	18.61	17.15

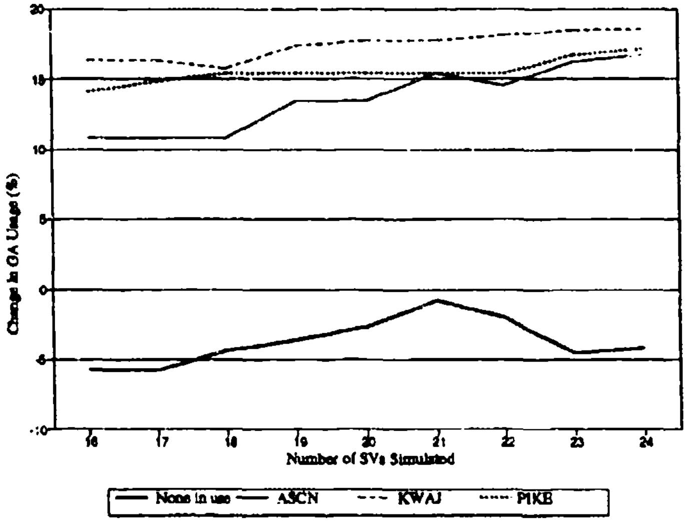

Figure 4.11. GA Utilization - CAPE and DIEG out.

The CAPE outage "forces" the reluctant PIKE to take a fair share of the load, which reduces the additional load on ASCN and KWAJ to reasonable levels.

Table 4.22. Individual GA Utilization with CAPE and KWAJ Out.

	Percentage of total simulation time						Change from "No Outage"			
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	ASCN Δ%	DIEG Δ%	PIKE Δ%
16	44.97	18.09	0.00	15.45	0.00	37.78	-5.62	1.87	1.94	34.93
17	40.42	20.03	0.00	15.45	0.00	40.76	-6.77	1.87	1.94	37.36
18	35.94	21.98	0.00	15.45	0.00	43.75	-6.24	1.88	1.94	40.35
19	32.15	23.72	0.00	18.65	0.00	43.75	-7.09	2.09	1.95	40.35
20	30.76	25.24	0.00	19.06	0.00	46.74	-5.77	2.08	1.94	43.34
21	26.74	27.47	0.00	21.01	0.00	48.26	-5.76	1.32	3.89	44.86
22	24.27	30.45	0.00	22.1[illegible]	0.00	49.44	-7.29	2.46	3.89	46.04
23	21.22	30.45	0.00	24.86	0.00	51.67	-10.13	2.46	4.655	47.50
24	20.00	32.01	0.00	24.86	0.00	55.42	-9.62	3.02	4.65	51.25

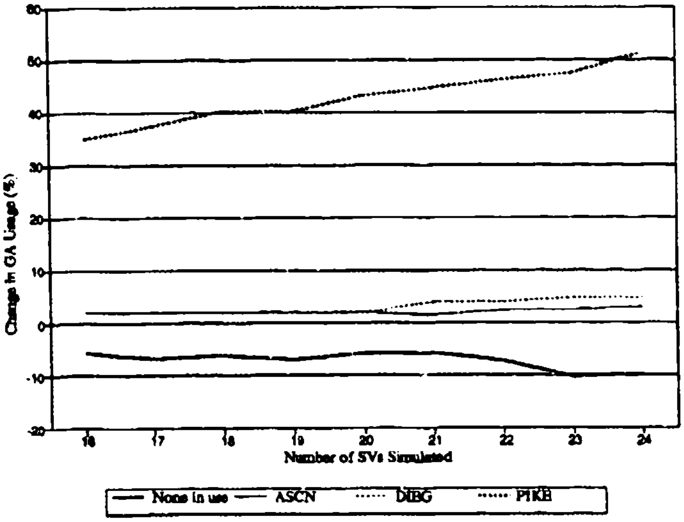

Figure 4.12. GA Utilization - CAPE and KWAJ out.

PIKE is the only alternative for scheduling activities for nearly half of the satellite's orbit. The system would have scheduling problems with this scenario if PIKE use was not close to zero in the baseline. As such, the system handles this setup better than any other dual outage, and better even than a DIEG-alone outage.

Table 4.23. Individual GA Utilization with D!EG and KWAJ Out.

	Percentage of total simulation time						Change from "No Outage"			
SVs	No GAs in use	ASCN in use	CAPE in use	DIEG in us	KWAJ in use	PIKE in use	No GAs Δ%	ASCN Δ%	CAPE Δ%	PIKE Δ%
16	47.36	30.07	30.83	0.00	0.00	9.10	-3.23	13.85	3.09	6.25
17	46.50	31.63	31.94	0.00	0.00	11.35	-0.69	13.47	3.47	7.95
18	41.28	32.64	32.88	0.00	0.00	14.34	-0.90	12.54	4.41	10.94
19	38.78	35.83	34.62	0.00	0.00	15.10	-0.46	14.20	5.94	11.70
20	38.02	37.40	34.83	0.00	0.00	16.26	1.49	14.24	6.15	14.86
21	33.96	41.56	35.59	0.00	0.00	19.03	1.46	15.41	6.91	15.63
22	31.42	44.55	35.59	0.00	0.00	20.97	-0.14	16.56	6.53	17.57
23	30.21	45.90	37.81	0.00	0.00	22.33	-1.14	17.91	7.29	18.16
24	27.85	46.32	37.81	0.00	0.00	27.99	-1.77	17.33	6.35	23.82

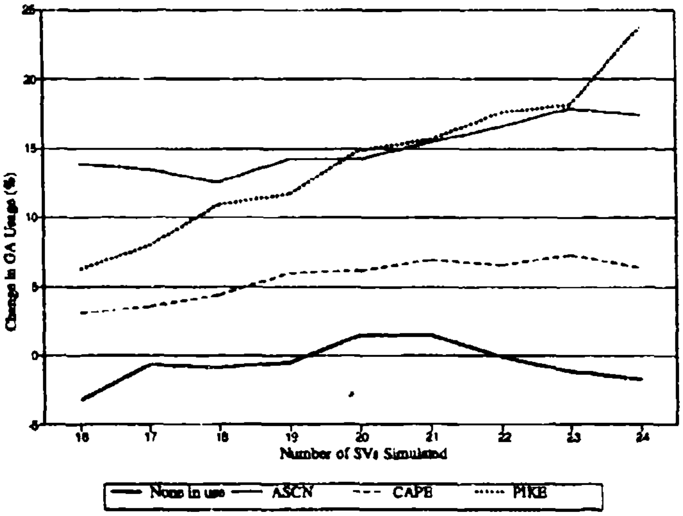

Figure 4.13. GA Utilization - DIEG and KWAJ out.

With PIKE, CAPE, and ASCN clustered into approximately 90o of longitude, this scenario opens the largest gap in the coverage. As indicated earlier, the system has the most failed supports and the highest tardiness while trying to compensate. ASCN is pushed to nearly the same utilization than with CAPE and DIEG out, while CAPE and PIKE stays under-utilized due to their westerly position relative to the outage.

Pages 4-34 through 4-37 show the percentage of total simulation time the system operates with zero, one, two, and three GAs in simultaneous use. Note that even with the simulated 24 satellite constellation, the are three GAs (and thus three Satellite Support Operators) in use only 8.68% of the time at most. Nearly 70% of the time there is less than two GAs in use. If the validity of the model could be improved to the extent these values could be assumed to reflect actual usage, they could be used by management to aid workload planning and personnel scheduling.

One interesting deviation in the otherwise consistent data is that with ASCN unavailable and the constellation size between 19 and 22, three GAs are in use more and one GA in use significantly less than in the other single-outage scenarios. It appears that with ASCN out the system the scheduler must cluster the activities slightly more than the other scenarios. This is also indicated by the higher "none-in-use" percentage during these same experiments. Figures 4.14 and 4.15 show this effect graphically. These graphs show the change in the number of GAs utilized for the ASCN and CAPE outage scenarios over the baseline no-outage condition (the CAPE outage scenario was chosen as a representative example). Both graphs compare the utilization for 21 simulated satellites. Note the higher number and wider range of GA utilization changes for the ASCN-out over the CAPE-out scenario, and the "clumping" of the ASCN-out changes at regular intervals (Figure 4.14).

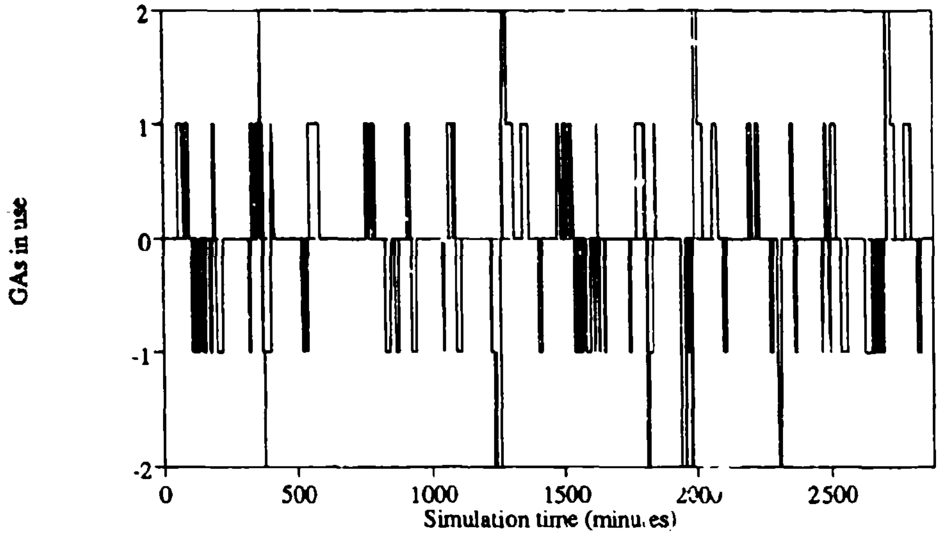

Figure 4.14. Change in GA Use due to ASCN Outage (21 SVs).

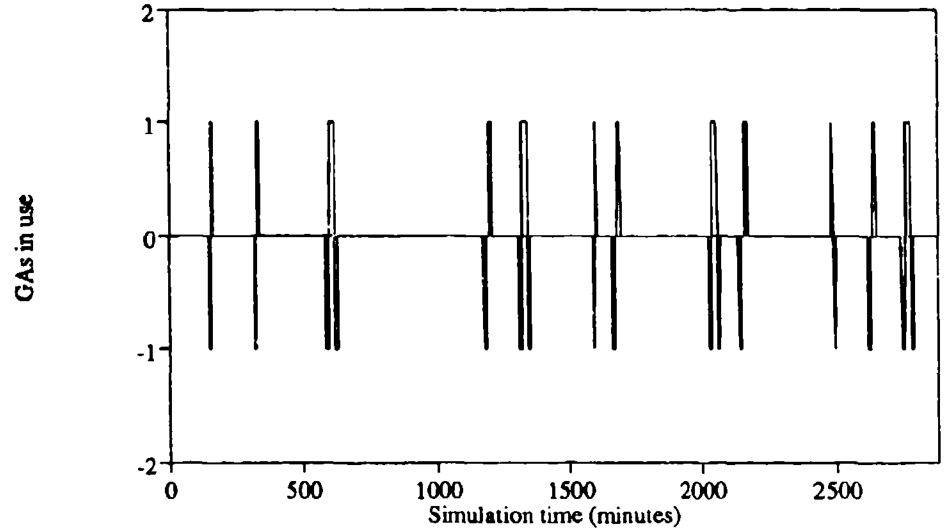

Figure 4.15. Change in GA Use due to CAPE Outage (21 SVs).

Table 4.24. Percent of Total Simulation Time with Zero GAs In Use.

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	[illegible]	48.47	46.98	46.11	48.33	47.60	46.35	42.60	43.51	47.36	44.79	44.97
17	[illegible].19	45.07	43.65	42.74	45.31	44.06	43.33	39.66	41.81	46.50	41.42	40.42
18	42.18	40.73	39.62	38.72	40.73	40 03	38.92	36.56	38.51	41.28	37.78	35.94
19	39.24	39.27	35.82	35.56	37.15	36.46	36.81	34.55	36.77	38.78	35.59	32.15
20	36.53	37.50	34.72	32.50	35.97	35.35	35.49	33.82	35.03	38.02	33.82	30.76
21	32.50	35.03	30.69	30.35	31.04	31.32	29.79	30.63	33.13	33.96	31.67	26.74
22	31.56	32.81	30.07	28.75	28.85	30.94	27.43	29.31	32.01	31.42	29.51	24.27
23	31.35	31.28	29.86	26.74	27 88	30.03	26.74	27.71	28.85	30.21	26.81	21.22
24	29.62	28.99	27.78	26.70	27.29	29.17	24.90	26.49	28.09	27.85	25.42	20.00

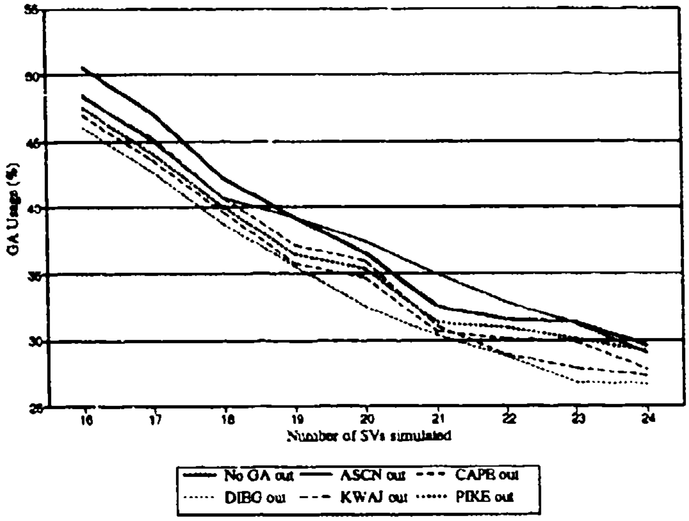

Figure 4.16. Percentage of Time No GAs in Simultaneous Use.

Table 4.25. Percent of Total Simulation Time with One GA In Use.

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	30.90	34.03	37.43	37.99	34.44	35.35	38.72	44.20	41.81	36.94	40.45	40.24
17	32.78	35.90	39.20	39.49	35.56	37.50	39.83	44.83	40.42	39.06	42.26	44.41
18	36.94	40.14	42.33	43.19	39.79	40.63	43.92	46.08	42.12	39.66	45.28	48.58
19	39.38	39.06	45.17	44.48	42.22	43.06	43.40	45.45	40.94	40.56	45.35	51.42
20	41.15	39.44	43.06	46.25	41.01	40.94	41.70	42.71	40.28	38.16	43.99	49.24
21	44.27	39.06	46.18	45.69	45.17	44.06	47.40	44.93	39.24	40.38	43.44	51.81
22	42.95	38.72	44.31	46.08	45.21	41.70	47.67	42.99	38.54	40.97	43.68	51.70
23	39.97	37.43	41.32	46.18	43.75	40.14	45.07	41.25	39.97	39.48	45.03	52.81
24	38.68	38.09	40.69	42.01	40.73	37.92	43.85	40.10	38.47	39.86	43.26	50.59

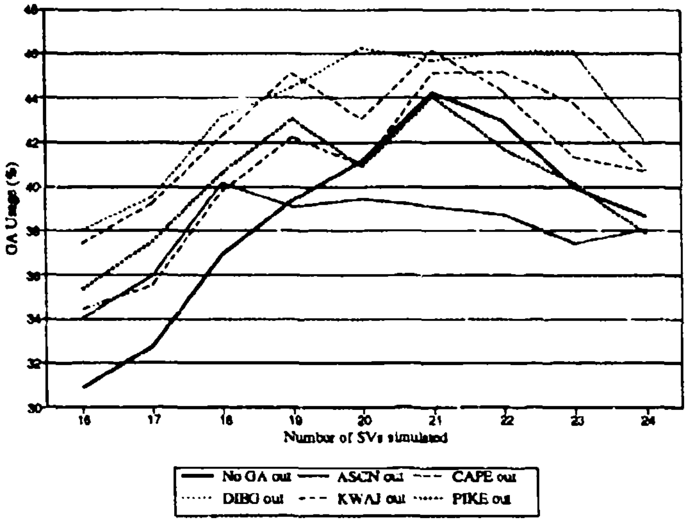

Figure 4.17. Percentage of Time One GAs in Simultaneous Use.

Table 4.26. Percent of Total Simulation Time with Two GAs In Use.

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	15.10	15.21	12.88	14.38	14.79	15.17	12.19	12.08	13.40	14.03	13.40	13.30
17	16.63	16.74	14.41	15.94	16.70	16.56	14.10	14.38	16.35	14.86	14.97	13.68
18	16.49	16.35	15.31	16.28	17.05	17.47	14.20	16.22	17.15	16.91	14.93	13.85
19	17.43	17.95	16.04	17.88	17.99	18.40	16.67	18.58	19.41	16.98	15.66	14.58
20	17.08	17.57	18.68	18.58	19.03	21.04	19.10	21.32	21.98	19.13	18.75	18.19
21	17.99	20.42	19.58	21.22	19.79	21.94	19.10	21.15	23.89	21.18	21.39	19.44
22	18.51	22.85	20.28	20.31	21.35	22.88	20.69	24.06	23.68	22.67	22.43	21.74
23	20.17	25.07	21.94	21.22	22.26	23.78	23.06	27.40	25.38	24.38	22.85	23.75
24	23.02	25.69	24.51	24.72	24.76	25.90	25.31	28.40	25.76	24.62	25.69	26.53

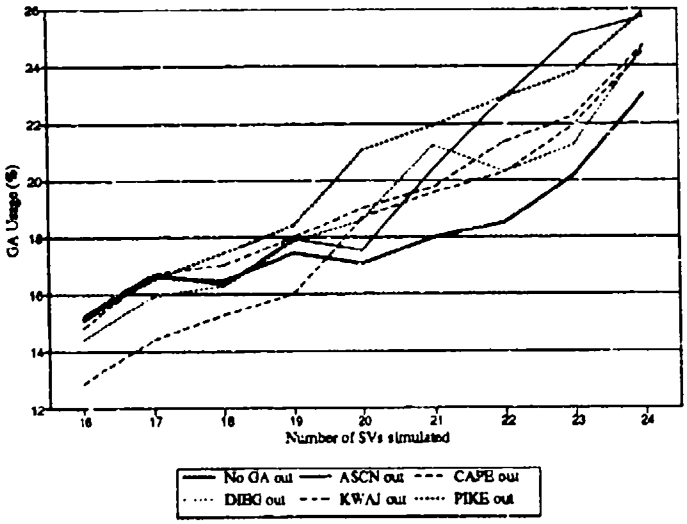

Figure 4.18. Percentage of Time Two GAs in Simultaneous Use.

Table 4.27. Percent of Total Simulation Time with Three GAs In Use.

	Ground Antenna(s) Unavailable											
SVS	NONE	ASCN	CAPE	DIEG	KWAJ	PIKE	A/K	A/C	A/D	K/D	C/D	C/K
16	3.40	2.29	2.71	1.53	2.43	1.88	2.74	1.11	1.28	1.67	1.35	1.49
17	3.40	2.29	2.74	1.53	2.43	1.88	2.74	1.15	1.42	2.05	1.35	1.49
18	3.75	2.78	2.74	1.81	2.43	1.88	2.95	1.15	2.22	2.12	2.01	1.63
19	3.96	3.72	2.95	2.08	2.64	2.08	3.13	1.42	2.88	3.68	3.40	1.84
20	5.24	5.49	3.54	2.67	3.99	2.67	3.72	2.15	3.51	4.69	3.44	1.81
21	5.24	5.49	3.54	2.74	3.99	2.67	3.72	3.30	3.75	4.48	3.51	2.01
22	6.95	5.63	5.35	4.86	4.58	4.48	4.20	3.65	5.76	4.03	4.38	2.29
23	8.51	6.22	6.88	5.87	6.11	6.04	5.14	3.65	5.80	5.94	5.28	2.22
24	8.68	7.22	7.01	6.56	7.22	7.01	5.94	5.00	7.67	7.67	5.63	2.88

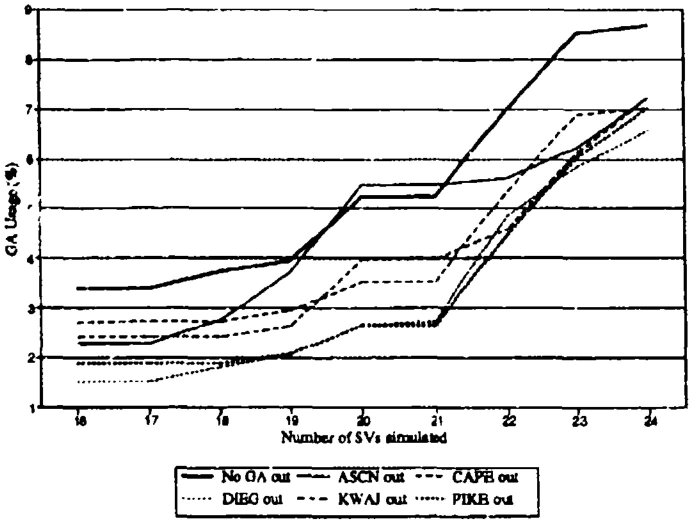

Figure 4.19. Percentage of Time Three GAs in Simultaneous Use.

4.4.2.4 Ground Antenna Utilization Results. As with the activity scheduling results, the Ground Antenna utilization rates under the varying experimental conditions confirm the proper operation of the MCS model. Mean utilization in the simulation is within 13% of the sample value for the real MCS GA utilization calculated from MCS logs. The simulated effects of single and dual GA outages on the utilization of the remaining Ground Antennas are consistently explained by the geographical arrangement of the GAs and their coverage areas.

The variations in the percentage of GAs in use for each scenario are consistent and logical. The results, along with crewmember loading studies, could be used to justify changes to crew manning, as an SSO and other crewmembers are "in use" to the same degree the GAs are in use. The results also add more weight to the conclusion that the model is functioning in a manner similar to the MCS and consistent with current MCS operations.

V. Conclusions/Recommendations.

5.1 Conclusions.

The ultimate source for the direction and implementation of this research was the problem of giving the Global Positioning System planners and managers a tool to describe how system growth could affect operational mission performance. Has this tool been created, and what is its efficacy in predicting MCS mission performance?

In the final analysis, the MCS operations simulator created during this research provides results comparable to the MCS in terms of schedule validity and operational effectiveness. It produced an valid satellite contact schedule for the full 24-satellite constellation, with any four of the five Ground Antennas available for use. With three GAs available, the scheduler failed to schedule at most four of 280 activities. The resulting schedule differed from the MCS-generated schedule due to its use of a subset of all the possible MCS activities and the rudimentary scheduling algorithm used in the simulation. While this difference precluded direct comparison of model results to MCS output, the schedule created was sufficiently close to a "real" schedule to cause the simulation GA utilization results to match the known response of the MCS to GA outages. With additional and more detailed rules for defining the scheduling process, the simulation could schedule even closer to reality. Of course, if such a sophisticated, rule-based scheduler was created (one that that produced *better* schedules than the MCS schedulers), the human element could be taken "out-of-the-loop" at the MCS and the operational and simulated schedules would be identical. An operational simulation under these conditions could predict MCS performance very accurately.

The MCS simulator created here has other useful aspects. The behavior of the scheduler was used to rate the relative significance of a number of MCS failure scenarios. This was done by assuming the human scheduler would have increased difficulty creating a schedule under the same conditions as the simulator. Using this gauge, the various combinations of GA outages were rated by impact to operations, with the combination of a KWAJ/DIEG outage being the most significant. Another indication was that a DIEG outage was more difficult to schedule than some combinations to two simultaneous GA outages. GA utilization rates produced by the simulation were explained by the geographical distribution of Ground Antennas and the effects of the outages on the world-wide coverage.

With few changes, this model could be adapted to simulate any satellite operational environment. The factors that make the model uniquely MCS-based are all contained in data files, not coded into the simulator program. In fact, with slight modification, the simulation could be used to model any operational environment that is driven by scheduled events with fixed time windows.

5.2 Recommendations.

The simulation as it stands is a major first step in creating a usable MCS operations simulation. It also provides clues to guide the creation of an artificial intelligence-based scheduler or scheduler's assistant. The following list describe recommendations for improvement of the MCS model and directions of further research in this subject.

5.2.1 Add Activities. Only a limited number of MCS activities were used in this model, under the assumption that the number used were sufficient to test model performance and perform initial experiments. There are other less-frequently scheduled activities, related to satellite eclipse operation, for instance, that need to be added to improve the fidelity of the model.

5.2.2 Extend Environment. The MCS can operate with as many as 30 GPS satellites simultaneously. In addition, if 30 satellites are required someday and the present ground station network proves to be incapable of handling the load, another GA might need to be added. Experiments could be run to test this hypothesis, and to aid in determining the optimum location for the additional GA.

5.2.3 GA Selection. The simulation currently steps through the list of GAs in fixed order when selecting the location for scheduling an activity. The next order of embellishment to the scheduling algorithm should include a smarter method for selecting GAs, either a random selection of equally valid choices or more rules to perform the selection"smartly".

5.2.4 Operations. There was insufficient time and cause to experiment the stochastic operations performance model written into the simulation. This was due to the conservative activity duration times associated with GPS operations. If this model is used to simulate other operations, or research into the effects of activity durations modeled as random variables, the operations simulation segment is there for that use.

Appendix A. *MCS Simulation SIMSCRIPT II.5 Program Listing*

A.1 *PREAMBLE*

```
''
''  MCS.SIM
''
''  Global Positioning System (GPS)
''  Master Control Station (MCS) Simulation
''
''  Capt David N. Koster
''  19 Oct 1992
''
''  PURPOSE:  This SIMSCRIPT II.5 program creates a valid
''   MCS (managed by the 2 satellite contact schedule, then
''   executes that schedule in a manner simulating the
''   performance by the MCS operations crew.
''

preamble
normally, mode is real
define minutes to mean units

''
'' This is the list of system queues.
''

the system owns
 a TO.BE.SCHEDULED,
 a SCHEDULED,
 a VIS.TABLE,
 a MASTER,
 a PERFORMED,
 a UNSCHEDULED

''
'' The only resource entity is used in the operations simulation
'' processes that immediately follow.
''

resources include
 CONTACT

processes include
 PERFORM.ACT,
 START.OPS

 every PERFORM.ACT has a THIS.ACT

 define
    THIS.ACT
    as a pointer variable

permanent entities include
 MASTER.ACT

 ''
 '' The MASTER.ACT entities act as a permanent, unchanged record
 '' of the activities required by each SV.  An iteration through
 '' MASTER.ACT is used when the various queues of temporary
 '' activities must be searched for one of each type of activity
 '' for each SV.
 ''

 every MASTER.ACT has
    a MNAME,        '' Support name
    a MSTART.TIME,        '' Start time (abs) of last performance
    a MLEAD.TIME,        '' Time until act due again (minutes)
    a MDURATION,        '' Duration of activity (minutes)
    a MINTERVAL,        '' Max interval between performance
    a MSTATUS,         '' Status of this activity
    a MSV,        '' SV name (launch number format)
    a MGA,        '' GA name
    a MGA.INDEX,        '' GA number (internal)
    a MBLOCK,        '' SV Block number (I or II)
    a MPRIORITY,        '' Activitiy priority
    a MVARIANCE,        '' Duration variance
    a MCRITICALITY        '' Importance to mission
    and may belong to
the MASTER

 define
    MSTART.TIME,
    MLEAD.TIME,
    MDURATION,
    MINTERVAL,
    MGA.INDEX,
    MBLOCK,
    MPRIORITY,
    MVARIANCE,
    MCRITICALITY
    as integer variables

  define
     MNAME,
     MSTATUS,
     MSV,
     MGA
     as text variables

 temporary entities
  include ACT and VIS.EVNT
```

```

''
'' The ACT entity is the building block for the scheduling routines
'' and the unit of performance during operations simulation. Each
'' entity is self-contained, as its attributes define the state
'' of the activity completely.  The attribute definitions are
'' identical to those of the MASTER.ACT, except for the dynamic
'' attributes which the MASTER.ACT entities do not use.
''

every ACT has
   a NAME,
   a START.TIME,
   a NEXT.START.TIME,     '' Tentative start of new activity
   a LEAD.TIME,
   a DURATION,
   a INTERVAL,
   a INTERVAL.OFFSET,     '' Additional interval time
   a STATUS,
   a SV,
   a GA,
   a GA.INDEX,
   a BLOCK,
   a PRIORITY,
   a VARIANCE,
   a CRITICALITY
   and may belong to
the TO.BE.SCHEDULED,
the SCHEDULED,
the PERFORMED,
the UNSCHEDULED

   define
START.TIME,
NEXT.START.TIME,
DURATION,
INTERVAL,
INTERVAL.OFFSET,
PRIORITY,
LEAD.TIME,
VARIANCE,
BLOCK,
GA.INDEX,
CRITICALITY
as integer variables

   define
NAME,
STATUS,
SV,
GA
```

```
as text variables

  ''
  '' The VIS.EVNT entities maintain the calculated rise and set
  '' times of SV/GA pairs.  The source of the data is external to
  '' this program.
  ''

  every VIS.EVNT has
     a SV,
     a GA,
     a GA.INDEX,
     a RISE.TIME,     '' Rise time of SV at GA
     a SET.TIME      '' Set time of SV at GA
     and may belong to
the VIS.TABLE

     define
RISE.TIME,
SET.TIME
as integer variables

 ''
 '' The following commands specify the order in which entities are
 '' maintained in the various system queues.  The rationale for
 '' indicated rankings are described in the routines where the queues
 '' are used.
 ''

 define
  TO.BE.SCHEDULED
  as a set ranked by
     high PRIORITY, then by
     low INTERVAL


 define
  SCHEDULED
  as a set ranked by
     low START.TIME, then by
     low SV

 define
  PERFORMED
  as a set ranked by
     high START.TIME, then by
     low SV

 define
  CHANGE.FLAG,        '' Indicates an exogenous parameter has changed
  CURRENT.TIME,        '' Minutes since 1 Jan 1992
```

```
EL.LIMIT,        '' Elevation threshold
END.DAY,       '' Desired simulation end time
END.HR,        ''       "
END.MIN,        ''       "
NUM.OF.GA,        '' Number of Ground Antennas used
NUM.OF.SV,        '' Number of Satellite Vehicles used
NUM.OF.OUTAGES,     '' Number of original (scheduled) GA outages
NUM.SIMUL.CONTACTS, '' Max number of simultaneous SV/GA contacts open
SIM.END.TIME,       '' Simulation completion time, jminutes
SIM.START.TIME,     '' Simulation start time, jminutes
START.DAY,        '' Desired simulation start time
START.HR,        ''      "
START.MIN,        ''      "
NMBR.OFFSET,        '' Variables for ACT interval offset statistics
MAX.OFFSET,        ''       "
MIN.OFFSET,        ''       "
OFFSET,        ''       "
NMBR.PRI,        '' Variables for ACT priority statistics
MAX.PRI,        ''       "
MIN.PRI,        ''       "
PRI,       ''       "
NMBR.UTIL,        '' Variables for GA utilization statistics
MAX.UTIL,        ''       "
MIN.UTIL,        ''       "
UTIL,        ''       "
NMBR.START.OFFSET,  '' Variables for validation offset statistics
MAX.START.OFFSET,   ''       "
MIN.START.OFFSET,   ''       "
START.OFFSET,       ''       "
RESV,        '' Variables for GA reservation statistics
MAX.PRIORITY,       '' Maximum ACT priority
INTERVAL.OFFSET.STEP, '' Value of interval offset increase
FAILED,        '' Number of failed supports
ASCN.UTIL,        '' Variables for individual GA statistics
CAPE.UTIL,        ''       "
DIEG.UTIL,        ''       "
KWAJ.UTIL,        ''       "
PIKE.UTIL,        ''       "
ASCN.RESV,        ''       "
CAPE.RESV,        ''       "
DIEG.RESV,        ''       "
KWAJ.RESV,        ''       "
PIKE.RESV       ''       "
as integer variables

define
 REMARK1,        '' Text description of experiment, parameters, etc.
 REMARK2,        '' Text description of experiment, parameters, etc.
 REMARK3,        '' Text description of experiment, parameters, etc.
 VIS.FILE,        '' Pointer to file containing visibility data
 TIME.FILE,        '' Pointer to file containing sim parameter data
 OUTAGE.FILE,       '' Pointer to file containing GA outage data
 ACT.FILE,       '' Pointer to file containing initial activity data
 OUT.FILE,       '' Pointer to file to contain sim results
 VAL.OUT       '' Pointer to file to contain validation results
 as text variables

''
'' The GA.USAGE arrays maintains the GA utilization data for each
'' GA for each minute of the simulation.
''

define
 GA.USAGE
 as a 2-dim text array

''
'' The following TALLY statements invoke autonomous system monitoring
'' of the variables named in the last line.  Each time the variables
'' change, the new value is automatically noted and the listed
'' statistics recalculated.
''

tally
 MN.OFFSET as the mean,
 STD.OFFSET as the std.dev,
 NMBR.OFFSET as the number,
 MAX.OFFSET as the maximum,
 MIN.OFFSET as the minimum,
 HIST.OFFSET(0 to 240 by 10) as the histogram
 of OFFSET

tally
 NMBR.UTIL as the number,
 MN.UTIL as the mean,
 STD.UTIL as the std.dev,
 MAX.UTIL as the maximum,
 MIN.UTIL as the minimum,
 HIST.UTIL(0 to 5 by 1) as the histogram
 of UTIL

tally
 NMBR.RESV as the number,
 MN.RESV as the mean,
 STD.RESV as the std.dev,
 MAX.RESV as the maximum,
 MIN.RESV as the minimum,
 HIST.RESV(0 to 5 by 1) as the histogram
 of RESV

tally
 NMBR.PRI as the number,
 MN.PRI as the mean,
 STD.PRI as the std.dev,
 MAX.PRI as the maximum,
 MIN.PRI as the minimum,
 HIST.PRI(0 to 11 by 1) as the histogram
 of PRI

tally
 NMBR.START.OFFSET as the number,
 MN.START.OFFSET as the mean,
 STD.START.OFFSET as the std.dev,
 MAX.START.OFFSET as the maximum,
 MIN.START.OFFSET as the minimum,
 HIST.SO(0 to 200 by 10) as the histogram
 of START.OFFSET

tally
 MN.AUTIL as the mean,
 STD.AUTIL as the std.dev
 of ASCN.UTIL

tally
 MN.CUTIL as the mean,
 STD.CUTIL as the std.dev
 of CAPE.UTIL

tally
 MN.DUTIL as the mean,
 STD.DUTIL as the std.dev
 of DIEG.UTIL

tally
 MN.KUTIL as the mean,
 STD.KUTIL as the std.dev
 of KWAJ.UTIL

tally
 MN.PUTIL as the mean,
 STD.PUTIL as the std.dev
 of PIKE.UTIL

tally
 MN.ARESV as the mean,
 STD.ARESV as the std.dev
 of ASCN.RESV

tally
 MN.CRESV as the mean,
 STD.CRESV as the std.dev
 of CAPE.RESV

```

```
tally
 MN.DRESV as the mean,
 STD.DRESV as the std.dev
 of DIEG.RESV

tally
 MN.KRESV as the mean,
 STD.KRESV as the std.dev
 of KWAJ.RESV

tally
 MN.PRESV as the mean,
 STD.PRESV as the std.dev
 of PIKE.RESV

end ''the preamble
```

A.2 MAIN

```

''**********************************************
''**********************************************
''**********************************************

main

''
'' The MAIN program segment control the flow of the program by
'' defining the execution sequence of the program routines and
'' processes.
''
'' The first four routines initialize the simulation and are only
'' executed once at the start of the simulation
''

now READ.DATA
now READ.VIS
now INIT.GA.USE
now INIT.ACTS

''
'' The PRESCHEDULE routine kicks off the scheduling sequence.  It
'' will be called anytime a new schedule is required.
''

now PRESCHEDULE

''
'' The START.OPS process is the first step in the operations
'' simulation process, which starts after the initial schedule has
'' been calculated.
''

schedule a START.OPS now
start simulation

''
'' The remainder of the routines called by MAIN describe the state
'' of the simulation after the simulation has completed, and
'' statistics pertaining to critical performance measures.
''

now REPORT.USE
now REPORT.QUEUES
now ANALYSIS
now VALIDATE
stop

end''main
```

A.3 READ.DATA Routine

```

''*********************************************
''*********************************************
''*********************************************

routine READ.DATA

''
'' This routine "bootstraps" the remainder of the initialization
'' sequence. It reads data from the system default data file, which
'' sets the basic simulation parameters and points to the subsequent
'' input data files. This "indirect" input method allows all
'' simulation parameters to be changed by modifying the default data
'' file only.
''

''
'' Where are the other data files?
''

read
 VIS.FILE,
 TIME.FILE,
 OUTAGE.FILE,
 ACT.FILE,
 OUT.FILE,
 VAL.OUT

''
'' These variables allow the user to describe the simulation run.
'' They are later included with the output files to "stamp" the
'' simulation operating conditions on the output.
''

read
 REMARK1,
 REMARK2,
 REMARK3
 as 3 T *

open unit 2 for input, name is TIME.FILE
use unit 2 for input

''
'' Read the simulation time period desired.
''

read
 START.DAY,
 START.HR,
 START.MIN,
 END.DAY,
 END.HR,
 END.MIN

''
'' Convert the desired time limits to "jminutes".
''

let SIM.START.TIME = START.DAY*1440 +
   START.HR*60 +
   START.MIN
let SIM.END.TIME = END.DAY*1440 +
 END.HR*60 +
 END.MIN
let CURRENT.TIME = SIM.START.TIME

''
'' Read the number of simultaneous contacts allowed, the maximum
'' activity priority, and the initial interval offset incrementation
'' value.
''

read
 NUM.SIMUL.CONTACTS,
 MAX.PRIORITY,
 INTERVAL.OFFSET.STEP

close unit 2

''
'' The operations simulations process uses resources to keep track
'' of GA utilization. These resources are created now.
''

create every CONTACT(1)
let u.CONTACT(1) = NUM.SIMUL.CONTACTS

end''READ.DATA
```

A.4 READ.VIS Routine

```

''*********************************************
''*********************************************
''*********************************************

routine READ.VIS

''
'' This routine is used to read the externally prepared visibility record.
'' The data must be synchronized with the simulation time period and the
'' preliminary activity data.  The visibility data is then used to
'' initialize VIS.EVNT entities and the VIS.TABLE set.
''

define
 .i,          '' Counter
 .j,             '' Counter
 .NUM.OF.EVNTS, '' The number of visibility periods to be read
 .RDAY,          '' The day of the month the SV rises at GA
 .RHOUR,         '' The hour of the month the SV rises at GA
 .RMIN,          '' The minute of the month the SV rises at GA
 .SDAY,          '' The day of the month the SV sets at GA
 .SHOUR,       '' The hour of the month the SV sets at GA
 .SMIN,          '' The minute of the month the SV sets at GA
 .GA.INDEX,   '' The numerical "tag" for this GA
 .JDAY.OFFSET   '' The number of days from 1 Jan to the 1st of
            ''   the month of the simulation
 as integer variables

define
 .SV,     '' The SV text name
 .GA     '' The GA text name
 as text variables

open unit 2 for input, name is VIS.FILE
use unit 2 for input

''
'' First, the number of Ground Antennas used in the simulation is read,
'' then the days from 1 Jan to the start of the month the simulation
'' begins.  Note that for simplicity, the simulation cannot "wrap"
'' around a month boundry.
''

''
'' Here is a sample of the data format for the visibility data file,
'' for reference.
''
''   5 90    "NUM.OF.GA,.JDAY.OFFSET"
''   ASCN 1    ".GA,.GA.INDEX"
''   BII-26   9 18  39  1   11    ".SV,.RDAY,.RHOUR,.RMIN,.SHOUR,.SMIN
'' ...
''   BII-06  14 23   4  5   42
''   END 0 0 0 0 0
''   CAPE 2
''   BII-04   9 18  17  1    0
'' ...
''   BI-010  14 23  56  4   50
''   END 0 0 0 0 0
''

read
 NUM.OF.GA,
 .JDAY.OFFSET

for .i = 1 to NUM.OF.GA do

 read
    .GA,
    .GA.INDEX

 let .SV = ""
 until .SV = "END" do

    read
.SV,
.RDAY,
.RHOUR,
.RMIN,
.SHOUR,
.SMIN

    if .SV ne "END"

create a VIS.EVNT

''
'' Make a correction to the set day if the event overlaps
'' 0000hrs.
''

if .RHOUR <= .SHOUR
   let .SDAY = .RDAY
else
   let .SDAY = .RDAY + 1
endif

''
'' Calculate the number of minutes from 0000 1 Jan (called
'' jminutes).
''
```

```

let RISE.TIME(VIS.EVNT) = 1440*(.RDAY + .JDAY.OFFSET) +
  60*.RHOUR +
  .RMIN
let SET.TIME(VIS.EVNT) = 1440*(.SDAY + .JDAY.OFFSET) +
 60*.SHOUR +
 .SMIN
let GA(VIS.EVNT) = .GA
let GA.INDEX(VIS.EVNT) = .GA.INDEX
let SV(VIS.EVNT) = .SV

''
'' Once the entity has its attributes set, it is filed in
'' the VIS.TABLE queue by low START.TIME, which amounts to
'' chronological order.
''

file this VIS.EVNT in VIS.TABLE

     endif
  loop
 loop

 close unit 2

 end ''routine READ.VIS
```

A.5 INIT.GA.USE Routine

```

''**********************************************
''**********************************************
''**********************************************

routine INIT.GA.USE

''
'' This routine reads in an initial ground antenna use schedule.  The
'' use could be from prescheduled contacts, scheduled GA maintenance,
'' MCS scheduled maintenance or testing, or, in the case of validation,
'' unscheduled outages that occured during the validation period.  An
'' array is used to reduce storage requirements and improved processing
'' time.
''

define
 .GA,
 .GA.INDEX,
 .START.TIME,
 .DURATION,
 .i,
 .j,
 .OFFSET '' Used to translate jminutes to array index
 as integer variables

define
 .CAUSE '' The reason for the outage.
 as a text variable


reserve
 GA.USAGE(*,*)
 as 5 by (SIM.END.TIME - SIM.START.TIME + 1)

open unit 2 for input, name is OUTAGE.FILE
use unit 2 for input

''
'' A sample outage file is shown below:
''
''   "ASCN out for entire simulation."   "REMARK1"
''   "DIEG out for entire simulation."   "REMARK2"
''   2   "NUM.OF.OUTAGES"
''   1 144000 2880 OUTAGE    ".GA.INDEX,.START.TIME,.DURATION, .CAUSE"
''   3 144000 2880 OUTAGE
''

read
 REMARK1,
  REMARK2
  as 2 T *

 read NUM.OF.OUTAGES

 if NUM.OF.OUTAGES > 0

  for .GA = 1 to NUM.OF.OUTAGES do

     read
.GA.INDEX,
.START.TIME,
.DURATION,
.CAUSE
     let .OFFSET = .START.TIME - SIM.START.TIME
     for .i = .OFFSET to .OFFSET + .DURATION do
let GA.USAGE(.GA.INDEX, .i+1) = .CAUSE
     loop
  loop

 endif

 close unit 2

 ''
 '' The following routine replaces the default vaules for the
 '' GA.USAGE array (a null string) with 10 spaces.  This improves
 '' the appearance of the array when output.
 ''

 for .i = 1 to 5 do
  for .j = 0 to SIM.END.TIME - SIM.START.TIME do
     if GA.USAGE(.i, .j+1) = ""
let GA.USAGE(.i, .j+1) = "          "
     endif
  loop
 loop

 end ''routine INIT.GA.USE
```

A.6 INIT.ACTS Routine

```

''************************************************
''************************************************
''************************************************

routine INIT.ACTS

''
'' This routine initializes the ACT entities needed to "hotstart" the
'' simulation.  They provides the necessary contact history (prior to
'' the start of the simulation) so the priority of the contacts can be
'' calculated.
''

define
 .SV,
 .BLOCK,
 .START.TIME,
 .DURATION,
 .INTERVAL,
 .PRIORITY,
 .VARIANCE,
 .CRITICALITY,
 .NUM.OF.ACTS
 as integer variables

define
 .SV.NAME,
 .ACT.NAME
 as text variables

open unit 2 for input, name is ACT.FILE
use unit 2 for input

''
'' Here is a sample ACT.FILE:
''
''  16     "NUM.OF.SV"
''  61     ".NUM.OF.ACTS"
''  BI-008 1 ADDKEYS 143815  10 1 1 4320
''  BI-008 1 NAV     143232   5 1 1 1440
''  BI-008 1 SOH     143805  10 1 1  480
''
''   ".SV.NAME, .BLOCK, .ACT.NAME, .START.TIME, .DURATION,
'' .VARIANCE, .CRITICALITY, .INTERVAL"
''
read
 NUM.OF.SV,
 .NUM.OF.ACTS

```

```
create every MASTER.ACT(.NUM.OF.ACTS)

for every MASTER.ACT do

 read
    .SV.NAME,
    .BLOCK,
    .ACT.NAME,
    .START.TIME,
    .DURATION,
    .VARIANCE,
    .CRITICALITY,
    .INTERVAL

 let MSV(MASTER.ACT) = .SV.NAME
 let MBLOCK(MASTER.ACT) = .BLOCK
 let MSTATUS(MASTER.ACT) = "UNSCHD"
 let MNAME(MASTER.ACT) = .ACT.NAME
 let MSTART.TIME(MASTER.ACT) = .START.TIME
 let MDURATION(MASTER.ACT) = .DURATION
 let MINTERVAL(MASTER.ACT) = .INTERVAL
 let MCRITICALITY(MASTER.ACT) = .CRITICALITY
 let MVARIANCE(MASTER.ACT) = .VARIANCE

 ''
 '' The ordering of the MASTER.ACT entities in MASTER is not
 '' important.
 ''

 file this MASTER.ACT in MASTER

 create an ACT

 let SV(ACT) = MSV(MASTER.ACT)
 let BLOCK(ACT) = MBLOCK(MASTER.ACT)
 let STATUS(ACT) = MSTATUS(MASTER.ACT)
 let NAME(ACT) = MNAME(MASTER.ACT)
 let LEAD.TIME(ACT) = MLEAD.TIME(MASTER.ACT)
 let START.TIME(ACT) = MSTART.TIME(MASTER.ACT)
 let DURATION(ACT) = MDURATION(MASTER.ACT)
 let INTERVAL(ACT) = MINTERVAL(MASTER.ACT)
 let CRITICALITY(ACT) = MCRITICALITY(MASTER.ACT)
 let VARIANCE(ACT) = MVARIANCE(HASTER.ACT)

 ''
 '' The ordering of the ACT entities in PERFORMED is by high
 '' START.TIME, so that when the queue is searched the latest
 '' example of each type of entity is found first.
 ''

 file this ACT in PERFORMED
```

```

loop

close unit 2

end'' routine INIT.ACTs
```

A.7 MAKE.NEW.ACT Routine

```

''*********************************************
''*********************************************
''*********************************************

routine MAKE.NEW.ACT given .ACT yielding .NEW.ACT

''
'' This routine simply takes the ACT entity whose pointer it is
'' passed and duplicates it, creating a new entity with the same
'' attribute values.  The pointer of the new entity is then
'' passed back to the calling routine.
''

define
 .NEW.ACT,
 .ACT
 as pointer variables

create an ACT called .NEW.ACT

let NAME(.NEW.ACT) = NAME(.ACT)
let SV(.NEW.ACT) = SV(.ACT)
let BLOCK(.NEW.ACT) = BLOCK(.ACT)
let START.TIME(.NEW.ACT) = START.TIME(.ACT)
let NEXT.START.TIME(.NEW.ACT) = NEXT.START.TIME(.ACT)
let DURATION(.NEW.ACT) = DURATION(.ACT)
let INTERVAL(.NEW.ACT) = INTERVAL(.ACT)
let INTERVAL.OFFSET(.NEW.ACT) = INTERVAL.OFFSET(.ACT)
let VARIANCE(.NEW.ACT) = VARIANCE(.ACT)
let CRITICALITY(.NEW.ACT) = CRITICALITY(.ACT)
let STATUS(.NEW.ACT) = STATUS(.ACT)
let PRIORITY(.NEW.ACT) = PRIORITY(.ACT)
let GA(.NEW.ACT) = GA(.ACT)
let GA.INDEX(.NEW.ACT) = GA.INDEX(.ACT)

end'' routine MAKE.NEW.ACT
```

A.8 PRESCHEDULE Routine

```

''*********************************************
''*********************************************
''*********************************************

routine PRESCHEDULE

''
'' FUNCTION:
''    - Create one ACT for each SV activity that needs to be performed
''      at least once more prior to the end of the simulation, and place
''      this ACT in TO.BE.SCHEDULED if there isn't one corresponding
''      to that activity in TO.BE SCHEDULED already.
''
'' INPUT CONDITIONS:
''    - One MASTER.ACT entity filed in MASTER for each routinely
''      required SV activity for each SV.
''
''    - One ACT entity filed in PERFORMED as above, with START.TIME(ACT)
''      indicating the last time this activity was performed prior
''      to the simulation start time.


define
 .NEW.ACT
 as pointer variables

print 1 line thus
PRESCHEDULE

''
'' Iterate through the list of activities, taking each ACT
'' one at a time.
''

for every MASTER.ACT in MASTER do

  ''
  '' Find the latest occurrance of the specific ACT in the
  '' PERFORMED queue.
  ''

  for every ACT in PERFORMED
with SV(ACT) = MSV(MASTER.ACT) and
     NAME(ACT) = MNAME(MASTER.ACT)
  find the first case
  if found

     ''
     '' This ACT need onlt be considered for further performance if
```

```
    '' it can be completed prior to the end of the simulation
    ''

    if START.TIME(ACT) +
  INTERVAL(ACT) +
  INTERVAL.OFFSET(ACT) +
  DURATION(ACT) < SIM.END.TIME

''
'' This ACT meets all the criteria for being scheduled again.
'' Make a copy, change the status to "UNSCHD", and file it in
'' TO.BE.SCHEDULED.  In addition, trigger the collection of
'' the performance statistics on this entity.
''

now MAKE.NEW.ACT giving ACT yielding .NEW.ACT
let STATUS(.NEW.ACT) = "UNSCHD"
let LEAD.TIME(.NEW.ACT) = START.TIME(.NEW.ACT) +
  INTERVAL(.NEW.ACT) -
  CURRENT.TIME
let OFFSET = INTERVAL.OFFSET(ACT)
let PRI = PRIORITY(ACT)
let PRIORITY(.NEW.ACT) = 0
let INTERVAL.OFFSET(.NEW.ACT) = 0
file this .NEW.ACT in TO.BE.SCHEDULED

     endif
  endif
 loop

 ''
 '' Do not call the scheduler if there are no activities to be scheduled.
 '' This routine falls through to the start of the simulation if the
 '' schedule is completed.  The tentatively scheduled activities are
 '' promoted to the SCHEDULED queue and the others are destroyed.
 ''

 if TO.BE.SCHEDULED is empty

  for every ACT in PERFORMED do
     if STATUS(ACT) = "TENT"

let STATUS(ACT) = "SCHD"
remove this ACT from PERFORMED
file this ACT in SCHEDULED

     else

remove this ACT from PERFORMED
destroy this ACT

```

```
    endif
 loop

else

 now SCHEDULE

endif

end ''PRESCHEDULE
```

A.9 SCHEDULE Routine

```


''*********************************************
''*********************************************
''*********************************************

routine SCHEDULE

''
'' This routine takes all the activity entities in the TO.BE.SCHEDULED
'' queue and iteratively tries to find suitable time periods in the
'' schedule for them to be performed.  If it succeeds, it calls
'' PRESCHEDULE to determine if there are further activities required.
'' If it cannot schedule one or more activities due to rescurce
'' it "forgets" what it has done to that point and restarts itself,
'' except that the troublesome activities have been modified to improve
'' the probability they will be scheduled.  If it exhausts all
'' possibility of scheduling an activity, the routine marks the
'' activity as hopeless and henceforth ignores it.

 define
  .CNCT.START,  '' Tentative start time currently under test
  .CONFLICT,  '' Flag indicating a scheduling conflict occurred
  .T,  '' Counter (in minutes)
  .i,  '' General purpose counter
  .N.CONTACTS,  '' Number of contacts currently scheduled
  .OFFSET,  '' Maps from sim time to GA.USAGE index
  .START.TIME,  '' Temporary storage for START.TIME calculations
  .GA,
  .CUT.OFF,  '' Earliest search time
  .RESCHD.FLAG,  '' Set if reschedule is required
  .PRIORITY.STEP '' Current priority step value
  as integer variables

 define
  .USE  '' temp storage for current GA.USAGE value
  as a text variable

 define
  .NEW.ACT
  as a pointer variable

 let .RESCHD.FLAG = 0
 let .PRIORITY.STEP = 1

 for every ACT in TO.BE.SCHEDULED
     with STATUS(ACT) = "UNSCHD"
     while STATUS(ACT) = "UNSCHD" do

  remove this ACT from TO.BE.SCHEDULED
  let .CUT.OFF = max.f(START.TIME(ACT), CURRENT.TIME)

  for .CNCT.START back from START.TIME(ACT) +
    INTERVAL(ACT) +
    INTERVAL.OFFSET(ACT) to .CUT.OFF
with .CNCT.START <= SIM.END.TIME
while STATUS(ACT) = "UNSCHD" do

     ''
     '' Step thru the visibility periods of the SV that's being
     '' scheduled at any (and all) GAs, provided the period of
     '' visibility is longer than the length of the desired contact
     ''

     for every VIS.EVNT in VIS.TABLE
   with SV(VIS.EVNT) = SV(ACT) and
RISE.TIME(VIS.EVNT) <= .CNCT.START and
SET.TIME(VIS.EVNT) >= .CNCT.START + DURATION(ACT)
   while STATUS(ACT) = "UNSCHD" do

''
'' Assume this prospective contact will not conflict
'' with another GA use
''

let .CONFLICT = 0

''
'' Step through the minutes of the prospective contact,
'' checking for correspondence in the GA use set.
'' If there is, set the conflict flag.
''

for .t = 0 to DURATION(ACT)
      while .CONFLICT ne 1 do

   let .OFFSET = (.CNCT.START + .t + 1) - SIM.START.TIME
   if GA.USAGE(GA.INDEX(VIS.EVNT), .OFFSET) ne "   "

      let .CONFLICT = 1

   endif

   let .N.CONTACTS = 0
   for .GA = 1 to 5
 with .GA ne GA.INDEX(VIS.EVNT)
 while .CONFLICT ne 1 do

      if substr.f(GA.USAGE(.GA,.OFFSET),1,6) = SV(ACT)
 let .CONFLICT = 1
      endif

      let .USE = GA.USAGE(.GA,.OFFSET)
      if substr.f(.USE,1,1) = "B"
 add 1 to .N.CONTACTS
      endif

   loop'' next .GA

   if .N.CONTACTS >= NUM.SIMUL.CONTACTS
      let .CONFLICT = 1
   endif

loop'' next .t

''
'' If this spot is reached with no conflict, the current ACT
'' is tentatively scheduled starting at .CNCT.START.
''

if .CONFLICT = 0

   let NEXT.START.TIME(ACT) = .CNCT.START
   let GA(ACT) = GA(VIS.EVNT)
   let GA.INDEX(ACT) = GA.INDEX(VIS.EVNT)
   let STATUS(ACT) = "TENT"

   ''
   '' create a GA.IN.USE event to prevent further
   '' use of this time slot
   ''

   let .OFFSET = .CNCT.START - SIM.START.TIME

   for i = .OFFSET to .OFFSET + DURATION(ACT) do
      let GA.USAGE(GA.INDEX(VIS.EVNT), .i+1) =
  concat.f(SV(ACT),"/",NAME(ACT))
   loop

endif '' if .CONFLICT = 0

     loop '' next visibility VIS.EVNT

  loop '' next .CNCT.START.TIME

  ''
  '' Any activity involved in a conflict will arrive here still UNSCHD.
  '' Either its PRIORITY or INTERVAL.OFFSET will be increased.
  ''

  if STATUS(ACT) = "UNSCHD"

```

```
     if PRIORITY(ACT) < MAX.PRIORITY

add .PRIORITY.STEP to PRIORITY(ACT)
add 1 to .PRIORITY.STEP
let STATUS(ACT) = "RESCHD"
let .RESCHD.FLAG = 1

     else

let STATUS(ACT) = "MISSED"
add INTERVAL.OFFSET.STEP to INTERVAL.OFFSET(ACT)
let .RESCHD.FLAG = 1

     endif '' PRIORITY(ACT) < MAX.PRIORITY

  endif '' STATUS is UNSCHD

  ''
  '' If a "conflicting" entity has just had its INTERVAL.OFFSET raised
  '' to one-half its default interval, the routine will mark the place
  '' in the schedule where the activity would have 'naturally" fallen
  '' (keeping its high priority and offset for the data colection
  '' process). It then increments the FAILED counter, and makes a
  '' new activity with default priority and offset.  This activity
  '' is placed in TO.BE.SCHEDULED, representing the next time this
  '' activity is needed after the FAILED one.
  ''

  if INTERVAL.OFFSET(ACT) <= 0.5 * INTERVAL(ACT)

     file this ACT in TO.BE.SCHEDULED

  else

     let STATUS(ACT) = "FAILED"
     let .i = START.TIME(ACT)
     let START.TIME(ACT) = .i + INTERVAL(ACT)
     add 1 to FAILED
     now MAKE.NEW.ACT giving ACT yielding .NEW.ACT
     subtract 1 from PRIORITY(.NEW.ACT)
     file this .NEW.ACT in PERFORMED
     let PRIORITY(ACT) = 0
     let INTERVAL.OFFSET(ACT) = 0
     file this ACT in TO.BE.SCHEDULED

  endif

 loop   ''for every ACT in TO.BE.SCHEDULED

 ''
 '' If every activity was scheduled, the new activities are updated
'' and moved to PERFORMED to form the basis for the next activity
'' selection process.  If not, this iteration of SCHEDULE is undone,
'' the activities made UNSCHD, then SCHEDULE called again.
''

if .RESCHD.FLAG = 0

 for every ACT in TO.BE.SCHEDULED do

    let STATUS(ACT) = "TENT"
    let START.TIME(ACT) = NEXT.START.TIME(ACT)
    remove this ACT from TO.BE.SCHEDULED
    file this ACT in PERFORMED

 loop

 now PRESCHEDULE

else

 for every ACT in TO.BE.SCHEDULED
with STATUS(ACT) = "TENT" do

    let .OFFSET = NEXT.START.TIME(ACT) - SIM.START.TIME
    for .i = .OFFSET to .OFFSET + DURATION(ACT) do
    let GA.USAGE(GA.INDEX(ACT),.i+1) = "    "
    loop

 loop

 for every ACT in TO.BE.SCHEDULED,
    let STATUS(ACT) = "UNSCHD"

 now SCHEDULE

endif

end ''routine SCHEDULE
```

A.10 START.OPS Process

```


''*********************************************
''*********************************************
''*********************************************

process START.OPS

''
'' This process calls itself every simulation minute.  When called,
'' it updates the CURRENT.TIME, checks for available CONTACT
'' resources and, if any are available, draws the next ACT from
'' the SCHEDULED queue andassigns it to a PERFORM.ACT process.
''

define
 .i
 as an integer variable

let CURRENT.TIME = SIM.START.TIME + time.v

if SCHEDULED is not empty

 if u.CONTACT(1) > 0

     for .i = 1 to u.CONTACT(1) do

for every ACT in SCHEDULED,
  find the first case

remove this ACT from SCHEDULED
activate a PERFORM.ACT giving ACT now

     loop
  endif
 endif

 if CURRENT.TIME < SIM.END.TIME

  schedule a START.OPS in 1 minutes

 endif

 end''START.OPS
```

A.11 PERFORM.ACT Process

```

''*********************************************
''*********************************************
''*********************************************

process PERFORM.ACT given THIS.ACT

'' This process takes the activity entity passed from
'' START.OPS and:
''     - grabs one of the CONTACT resources,
''     - adds variation to the activity, if required,
''     - tests to ensure the activity can be performed
''       as scheduled,
''     - simulates the performance of the valid activities, and
''     - calls for a new schedule if an activity cannot be
''       performed.
''
define
 .i,
 .OFFSET,  '' Minutes into simulation
 .DURATION,
 .OFF.SCHEDULE  '' Set to one if activity cannot be performed
 as integer variables

define
 .USE
 as a text variable

define
 THIS.ACT
 as a pointer variable

request 1 CONTACT(1)

''
'' These statements are modified if activity duration variance
'' is desired.
''

let .DURATION = DURATION(THIS.ACT)
let DURATION(THIS.ACT) = .DURATION

let .OFF.SCHEDULE = 0

''
'' It is good if the activity start time equals the current sim time.
'' Test further and take corrective action if they are not equal.
''

if START.TIME(THIS.ACT) ne CURRENT.TIME
```

```

  ''
  '' If the start time if later than the current time, the process
  '' halts until time catches up.  If earlier, check the visibility
  '' table and GA use for the possibility the activity cannot be
  '' performed now.  If it cannot, set the .OFF.SCHEDULE flag.
  ''

  if START.TIME(THIS.ACT) > CURRENT.TIME

     wait START.TIME(THIS.ACT) - CURRENT.TIME + 1 minutes

  else

     let .OFFSET = CURRENT.TIME - SIM.START.TIME

     for .i = .OFFSET to .OFFSET + DURATION(THIS.ACT)
   while .OFF.SCHEDULE = 0 do

for every VIS.EVNT in VIS.TABLE
      with GA(VIS.EVNT) = GA(THIS.ACT) and
   SV(VIS.EVNT) = SV(THIS.ACT) and
   RISE.TIME(VIS.EVNT) >= CURRENT.TIME + (.i - .OFFSET) and
   SET.TIME(VIS.EVNT) <= CURRENT.TIME + (.i - .OFFSET)
find the first case
if none

   let .OFF.SCHEDULE = 1

endif

let .USE = GA.USAGE(GA.INDEX(THIS.ACT),.i+1)
if .USE ne "      " and .USE ne SV(THIS.ACT)

   let .OFF.SCHEDULE = 1

endif
     loop
  endif
 endif

 if .OFF.SCHEDULE = 0

  ''
  '' The activity can be performed now...hold on to the resource
  '' for the duration of the activity, then file the activity in
  '' PERFORMED.
  ''

  let START.TIME(THIS.ACT) = CURRENT.TIME
  let STATUS(THIS.ACT) = "PERF"

  wait DURATION(THIS.ACT) minutes
  relinquish 1 CONTACT(1)
  file THIS.ACT in PERFORMED

 else

  ''
  '' Remove the the scheduled activities from SCHEDULED, clear
  '' their GA reservations from GA.USAGE, give up the CONTACT
  '' token, and call PRESCHEDULE to create a new schedule.
  ''

  for every ACT in SCHEDULED
with START.TIME(ACT) >= CURRENT.TIME do

     remove this ACT from SCHEDULED

     let .OFFSET = START.TIME(ACT) - SIM.START.TIME

     for .i = .OFFSET to .OFFSET + DURATION(ACT) do
let GA.USAGE(GA.INDEX(ACT),.i+1) = "       "
     loop

     destroy this ACT

  loop

  relinquish 1 CONTACT(1)
  now PRESCHEDULE

 endif


 end''PERFORM.ACT
```

A.12 REPORT.QUEUES Routine

```

''*********************************************
''*********************************************
''*********************************************

routine REPORT.QUEUES

''
'' This routine prints the contents of the systems queues.
''

define
 .i,
 .j
 as integer variables

print 1 line with CURRENT.TIME thus
REPORT QUEUES:  Current time = ******

if SCHEDULED is not empty
print 2 lines thus
SCHEDULED:
START     (NEXT) SV/ACT   DUR/INT    PRI/OFFSET   GA
for each ACT in SCHEDULED do
 print 1 line with
    trunc.f(START.TIME(ACT)/1440),
    trunc.f(frac.f(START.TIME(ACT)/1440)*24),
    mod.f(mod.f(START.TIME(ACT),1440),60),
    trunc.f(NEXT.START.TIME(ACT)/1440),
    trunc.f(frac.f(NEXT.START.TIME(ACT)/1440)*24),
    mod.f(mod.f(NEXT.START.TIME(ACT),1440),60),
    SV(ACT),
    NAME(ACT),
    DURATION(ACT),
    INTERVAL(ACT),
    PRIORITY(ACT),
    INTERVAL.OFFSET(ACT),
    GA(ACT) thus
***/**:** (***/**:**)  ***********/******  **/*****  **/***   *****
loop
endif

if TO.BE.SCHEDULED is not empty
print 2 lines thus
TO.BE.SCHEDULED:
START     (NEXT) SV/ACT   DUR/INT    PRI/OFFSET   GA

for each ACT in TO.BE.SCHEDULED do
 print 1 line with
    trunc.f(START.TIME(ACT)/1440),
      trunc.f(frac.f(START.TIME(ACT)/1440)*24),
      mod.f(mod.f(START.TIME(ACT),1440),60),
      trunc.f(NEXT.START.TIME(ACT)/1440),
      trunc.f(frac.f(NEXT.START.TIME(ACT)/1440)*24),
      mod.f(mod.f(NEXT.START.TIME(ACT),1440),60),
      SV(ACT),
      NAME(ACT),
      DURATION(ACT),
      INTERVAL(ACT),
      PRIORITY(ACT),
      INTERVAL.OFFSET(ACT),
      GA(ACT) thus
***/**:** (***/**:**) ***********/****** **/***** **/***   *****
  loop
  endif

  if PERFORMED is not empty
  print 2 lines thus
PERFORMED:
START     (NEXT) SV/ACT   DUR/INT    PRI/OFFSET   GA
  for each ACT in PERFORMED do
   print 1 line with
      trunc.f(START.TIME(ACT)/1440),
      trunc.f(frac.f(START.TIME(ACT)/1440)*24),
      mod.f(mod.f(START.TIME(ACT),1440),60),
      trunc.f(NEXT.START.TIME(ACT)/1440),
      trunc.f(frac.f(NEXT.START.TIME(ACT)/1440)*24),
      mod.f(mod.f(NEXT.START.TIME(ACT),1440),60),
      SV(ACT),
      NAME(ACT),
      DURATION(ACT),
      INTERVAL(ACT),
      PRIORITY(ACT),
      INTERVAL.OFFSET(ACT),
      GA(ACT) thus
***/**:** (***/**:**) ***********/****** **/***** **/***   *****
  loop
  endif

  if UNSCHEDULED is not empty
  print 2 lines thus
UNSCHEDULED:
START     (NEXT) SV/ACT   DUR/INT PRI/INT.OFFSET
  for each ACT in UNSCHEDULED do
   print 1 line vith
      trunc.f(START.TIME(ACT)/1440),
      trunc.f(frac.f(START.TIME(ACT)/1440)*24),
      mod.f(mod.f(START.TIME(ACT),1440),60),
      trunc.f(NEXT.START.TIME(ACT)/1440),
      trunc.f(frac.f(NEXT.START.TIME(ACT)/1440)*24),
      mod.f(mod.f(NEXT.START.TIME(ACT),1440),60),
    SV(ACT),
    NAME(ACT),
    DURATION(ACT),
    INTERVAL(ACT),
    PRIORITY(ACT),
    INTERVAL.OFFSET(ACT) thus
***/**:** (***/**:**)  ***********/******  **/***** **/***
loop
endif

end''REPORT.QUEUES
```

A.13 REPORT.USE Routine

```

''*******************************************
''*******************************************
''*******************************************

routine REPORT.USE

''
'' This routine prints the GA use array to the standard
'' output file. Minutes if no GA in use are not printed.
''

define
 .i,
 .j
 as integer variables

print 1 line with CURRENT.TIME thus
REPORT USE:   Current time = ******

for .i = 1 to SIM.END.TIME - SIM.START.TIME  do
 let .j = .i + 144000

 if GA.USAGE(1,.i) ne "  " or
   GA.USAGE(2,.i) ne "  " or
   GA.USAGE(3,.i) ne "  " or
   GA.USAGE(4,.i) ne "  " or
   GA.USAGE(5,.i) ne "  "

   print 1 line with
trunc.f(.j/1440),
trunc.f(frac.f(.j/1440)*24),
mod.f(mod.f(.j,1440),60),
GA.USAGE(1,.i),
GA.USAGE(2,.i),
GA.USAGE(3,.i),
GA.USAGE(4,.i),
GA.USAGE(5,.i) thus
***/**:**    **********  **********  **********  **********  **********
 endif
loop

end''REPORT.USE
```

A.14 REPORT.VIS Routine

```

''**********************************************
'' *********************************************
''*****************    ************************

routine REPORT.VIS

''
'' This routine prints the visibility table to the standard
'' output file.
''

print 1 line with CURRENT.TIME thus
REPORT VIS:   Current time = ******

for each VIS.EVNT in VIS.TABLE do
 print 1 line with
     trunc.f(RISE.TIME(VIS.EVNT)/1440),
     trunc.f(frac.f(RISE.TIME(VIS.EVNT)/1440)*24),
     trunc.f(frac.f(frac.f(RISE.TIME(VIS.EVNT)/1440)*24)*60),
     trunc.f(SET.TIME(VIS.EVNT)/1440),
     trunc.f(frac.f(SET.TIME(VIS.EVNT)/1440)*24),
     trunc.f(frac.f(frac.f(SET.TIME(VIS.EVNT)/1440)*24)*60),
     RISE.TIME(VIS.EVNT),
     SET.TIME(VIS.EVNT),
     GA(VIS.EVNT),
     GA.INDEX(VIS.EVNT),
     SV(VIS.EVNT) thus
***/**:**  ***/**:**  (*******-******) ****(*)  ******
loop

end '' REPORT.VIS
```

A.15 ANALYSIS Routine

```

''*********************************************
''*********************************************
''*********************************************

routine ANALYSIS

''
'' This routine princ the simulation conditions, parameters, and
'' summarizes the results in terms of the predefined
'' performance measures.  The output is saved to the filename
'' defined by OUT.FILE.
''

define
 .i,
 .GA.INDEX,
 .START.TIME,
 .DURATION,
 .j,
 .RESV.TEMP,
 .UTIL.TEMP
 as integer variables

define
 .CAUSE
 as a text variable

open unit 3 for outpuc,
 name is OUT.FILE
use unit 3 for output

''
'' This crude but effective rcutine scans the GA.USE array and
'' results in utilization statistics for individual and total
'' GA use.
''

for .i = 1 to SIM.END.TIME - SIM.START.TIME  do

 let .UTIL.TEMP = 0
 let .RESV.TEMP = 0
 if substr.f(GA.USACE(1,.i),1,1) = 'B"
    let ASCN.UTIL = 1
    add 1 to .UTIL.TEMP
 else
    let ASCN.UTIL = 0
 endif
 if substr.f(GA.USAGE(1,.i),1,1) ne " "
    let ASCN.RESV = 1
```

```
   add 1 to .RESV.TEMP
else
   let ASCN.RESV = 0
endif
if substr.f(GA.USAGE(2,.i),1,1) = "B"
   let CAPE.UTIL = 1
   add 1 to .UTIL.TEMP
else
   let CAPE.UTIL = 0
endif
if substr.f(GA.USAGE(2,.i),1,1) ne " "
   let CAPE.RESV = 1
   add 1 to .RESV.TEMP
else
   let CAPE.RESV = 0
endif
if substr.f(GA.USAGE(3,.i),1,1) = "B"
   let DIEG.UTIL = 1
   add 1 to .UTIL.TEMP
else
   let DIEG.UTIL = 0
endif
if substr.f(GA.USAGE(3,.i),1,1) ne " "
   let DIEG.RESV = 1
   add 1 to .RESV.TEMP
else
   let DIEG.RESV = 0
endif
if substr.f(GA.USAGE(4,.i),1,1) = "B"
   let KWAJ.UTIL = 1
   add 1 to .UTIL.TEMP
else
   let KWAJ.UTIL = 0
endif
if substr.f(GA.USAGE(4,.i),1,1) ne " "
   let KWAJ.RESV = 1
   add 1 to .RESV.TEMP
else
   let KWAJ.RESV = 0
endif
if substr.f(GA.USAGE(5,.i),1,1) = "B"
   let PIKE.UTIL = 1
   add 1 to .UTIL.TEMP
else
   let PIKE.UTIL = 0
endif
if substr.f(GA.USAGE(5,.i),1,1) ne " "
   let PIKE.RESV = 1
   add 1 to .RESV.TEMP
else
   let PIKE.RESV = 0
```

```
       endif
       let UTIL = .UTIL.TEMP
       let RESV = .RESV.TEMP
       print 1 line with
          .i,
          .UTIL.TEMP thus
****** *

      loop

      print 10 lines with
       START.DAY,
       START.HR,
       START.MIN,
       END.DAY,
       END.HR,
       END.MIN,
       SIM.START.TIME,
       SIM.END.TIME,
       NUM.OF.SV,
       NUM.OF.GA,
       NUM.SIMUL.CONTACTS,
       REMARK1,
       REMARK2,
       REMARK3 thus

SCHEDULING INPUTS
TIME: ***/**:** to ***/**:** (****** to ******)
** SVs
** GAs
* Max contacts
************************************************************
************************************************************
************************************************************

      open unit 2 for input,
       name is OUTAGE.FILE

      use unit 2 for input

      read
       REMARK1,
       REMARK2
       as 2 T *

      read .i

      print 1 line with NUM.OF.OUTAGES thus
** Prescheduled outages:

      if NUM.OF.OUTAGES > 0
```

```

print 2 lines thus
Index    Start   Duration   Cause

for .i = 1 to NUM.OF.OUTAGES do

    read
.GA.INDEX,
.START.TIME,
.DURATION,
.CAUSE

    print 1 line with
.GA.INDEX,
.START.TIME,
.DURATION,
.CAUSE thus
*   ******     ****     ********************

 loop
endif

close unit 2

print 4 lines thus

RESULTS:
   Number    Max       Min   Mean    Std Dev

print 6 lines with
 NMBR.OFFSET,
 MAX.OFFSET,
 MIN.OFFSET,
 MN.OFFSET,
 STD.OFFSET,
 NMBR.PRI,
 MAX.PRI,
 MIN.PRI,
 MN.PRI,
 STD.PRI,
 NMBR.UTIL,
 MAX.UTIL,
 MIN.UTIL,
 MN.UTIL,
 STD.UTIL,
 NMBR.RESV,
 MAX.RESV,
 MIN.RESV,
 MN.RESV,
 STD.RESV,
 FAILED thus
```

```
OFFSET:      *****    ****     **** **.****    **.****
PRIORITY:    *****    ****     **** **.****    **.****
GA UTIL:     *****    ****     **** **.****    **.****
GA RESV:     *****    ****     **** **.****    **.****
FAILED SUPPORTS: **

print 4 lines thus

OFFSET:
Value  Number  Percent

for .i = 1 to 25 do
 print 1 line with
    (.i-1)*10,
    HIST.OFFSET(.i),
    100*HIST.OFFSET(.i)/NMBR.OFFSET thus
*   ****    ***.**
loop

print 4 lines thus

PRIORITY:
Value  Number  Percent

for .i = 1 to 12 do
 print 1 line with
    .i-1,
    HIST.PRI(.i),
    100*HIST.PRI(.i)/NMBR.PRI thus
*    ****    ***.**
loop

print 10 lines with
 MN.AUTIL*100,
 MN.ARESV*100,
 STD.AUTIL*100,
 STD.ARESV*100,
 MN.CUTIL*100,
 MN.CRESV*100,
 STD.CUTIL*100,
 STD.CRESV*100,
 MN.DUTIL*100,
 MN.DRESV*100,
 STD.DUTIL*100,
 STD.DRESV*100,
 MN.KUTIL*100,
 MN.KRESV*100,
 STD.KUTIL*100,
 STD.KRESV*100,
 MN.PUTIL*100,
 MN.PRESV*100,
STD.PUTIL*100,
STD.PRESV*100 thus

IIDIVIDUAL GA UTILIZATIOI(SV)/RESERVATIOI(ALL):
GA % used/resv  std used/resv

ASCI **.**/**.**  **.**/**.**
CAPE **.**/**.**  **.**/**.**
DIEG **.**/**.**  **.**/**.**
KWAJ **.**/**.**  **.**/**.**
PIKE **.**/**.**  **.**/**.**

print 4 lines thus

GA UTILIZATIOI(SV)/RESERVATIOI(ALL):
Value  Iumber  Percent

for .i = 1 to 6 do
 print 1 line with
     .i-1,
     HIST.UTIL(.i),
     HIST.RESV(.i),
     100*HIST.UTIL(.i)/IMBR.UTIL,
     100*HIST.RESV(.i)/IMBR.RESV thus
*  ****/****     ***.**/***.**
loop

close unit 3

end''AIALYSIS
```

A.16 VALIDATE Routine

```


''********************************************
''********************************************
''********************************************

routine VALIDATE

''
'' This routine is used only during the simulation validation step,
'' to provide quantative measure of the ability of the simulation to
'' duplicate the MCS scheduling results.  The file "valact.dat"
'' contains a set of "real" MCS-scheduled activities for the
'' validation period; the validation results are written to the
'' file pointed to by VAL.OUT.
''

define
 .i,
 .START.TIME,
 .DURATION,
 .BLOCK,
 .INTERVAL,
 .PRIORITY,
 .VARIANCE,
 .CRITICALITY,
 .GA.FLAG,
 .NUM.OF.VALACTS
 as integer variables

define
 .CAUSE,
 .GA,
 .SV.NAME,
 .ACT.NAME
 as text variables

open unit 2 for input,
 name is "valact.dat"
use unit 2 for input

open unit 3 for output,
 name is VAL.OUT
use unit 3 for output

''
'' Here is a excerpt from a "valact.dat" file:
''    57       ".NUM.OF.VALACTS"
''    BI-008 1 NAV      144660 5 1 1    1440 CAPE
''    BI-008 1 SOH      144200  10 1 1     480 DIEG
''    BI-009 1 NAV      144145 5 1 1    1440 CAPE
''       .SV.NAME, .BLOCK, .ACT.NAME, .START.TIME, .DURATION,
''       .VARIANCE, .CRITICALITY, .INTERVAL, .GA
''

read .NUM.OF.VALACTS
for .i = 1 to .NUM.OF.VALACTS do

 read
    .SV.NAME,
    .BLOCK,
    .ACT.NAME,
    .START.TIME,
    .DURATION,
    .VARIANCE,
    .CRITICALITY,
    .INTERVAL,
    .GA

 ''
 '' The first activity entity in the SCHEDULED queue that
 '' matches the current validation activity is referenced.
 '' the difference between the two start times is found and
 '' a counter is incremented if the system scheduled the same
 '' GA that was used in reality.  A TALLY statement is tracking
 '' value of the START.OFFSET.
 ''

 for every ACT in SCHEDULED
with SV(ACT) = .SV.NAME and
    NAME(ACT) = .ACT.NAME and
    START.TIME(ACT) >= SIM.START.TIME
 find the first case
 if found

    let START.OFFSET = abs.f(.START.TIME - START.TIME(ACT))
    if .GA = GA(ACT)
add 1 to .GA.FLAG
    endif
 endif
loop

print 10 lines with
 START.DAY,
 START.HR,
 START.MIN,
 END.DAY,
 END.HR,
 END.MIN,
 SIM.START.TIME,
 SIM.END.TIME,
NUM.OF.SV,
NUM.OF.GA,
NUM.SIMUL.CONTACTS,
REMARK1,
REMARK2,
REMARK3 thus

SCHEDULING INPUTS
TIME: ***/**:** to ***/**:** (****** to ******)
** SVs
** GAs
* Max contacts
**************************************************************
**************************************************************
**************************************************************

print 4 lines thus

VALIDATION RESULTS:
Number   Max     Min        Mean Std Dev

print 4 lines with
 NMBR.START.OFFSET,
 MAX.START.OFFSET,
 MIN.START.OFFSET,
 MN.START.OFFSET,
 STD.START.OFFSET,
 .GA.FLAG thus

START OFFSET:  ***** ****  ****    **.****    **.****
GA MATCH:      *****

print 4 lines thus

ACTIVITY START OFFSET:
Value  Number  Percent

for .i = 1 to 21 do
 print 1 line with
    (.i-1)*10,
    HIST.SO(.i),
    100*HIST.SO(.i)/NMBR.START.OFFSET thus
*  ****   ***.**
loop

close unit 2
close unit 3

end''VALIDATE
```

Appendix B. *MCS Simulation Example Input Data.*

B.1 Overview.

This appendix describes the input data used during a typical experimental run of the simulation program. As the total experimental suite was 108 runs, with each set of input data different for each run, it is not practical to provide the complete input data array here. However, the input data modification process required to duplicate the experimental results is described in Chapter 4. The purpose and use of the various data is described in Chapter 3 and the comments contained in the MCS.SIM program listing, Appendix A. General notes are provided with the listing of each data set.

B.2 Simulation VAX Command File.

This is not input data *per se*, but one of the command files used to direct the execution of the simulation runs. The MCS.SIM program was executed on the Air Force Institute of Technology Digital Equipment Corporation VAX 6420 (9:8.1). The default method for submitting SIMSCRIPT programs on this system does not easily allow batch submissions, so the following simple command file was created to speed the experiment process. Each input file was prepared in advance to specify the appropriate input and output files.

```
$ define/user sys$input s16.dat
$ define/user sys$output s16.out
$ run mcs
$ define/user sys$input s17.dat
$ define/user sys$output s17.out
$ run mcs
$ define/user sys$input s18.dat
$ define/user sys$output s18.out
$ run mcs
$ define/user sys$input s19.dat
$ define/user sys$output s19.out
$ run mcs
$ define/user sys$input s20.dat
$ define/user sys$output s20.out
$ run mcs
$ define/user sys$input s21.dat
$ define/user sys$output s21.out
$ run mcs
$ define/user sys$input s22.dat
$ define/user sys$output s22.out
$ run mcs
$ define/user sys$input s23.dat
$ define/user sys$output s23.out
$ run mcs
$ define/user sys$input s24.dat
$ define/user sys$output s24.out
$ run mcs
```

B.3 Simulation Parameter Data

The following data is maintained in the SIMSCRIPT system default input datafile. For the VAX implementation of SIMSCRIPT, this is the file specified by the user in response to the "Enter name of Input File (.DAT assumed) (ie. DATA1)" request of the RUNSIM start program. When using the command file listed in Section B.2, the SIMSCRIPT executive will read the data file defined as the default system input (e.g., "s24.dat").

```
visall.dat
time.dat
outage.dat
act24.dat
r24.out
val.out
"No GA maint - no j100 CPU outage"
"Pike full availability"
"24 SVs"
```

B.4 Visibility Data

This data describes the visibility periods between satellite vehicle and ground antenna pairs. The data used for the validation and testing of this MCS Simulator was created by the program

Pass Scheduler (6). A description of *Pass Scheduler* and the process used to create the data below is described in Appendix D.

```
5 90
ASCN 1
BII-26         9      18      39       1      11
BII-09         9      19       5       1       9
BII-27         9      20      32       3      11
BII-01         9      20      43       3      16
BII-13         9      21       3       2      32
BII-04         9      22      14       7      2[illegible]
BI-008         9      22      38  [illegible] 54
BII-06         9      23      24       6       3
BII-02        10       0      39       9      52
BII-29        10       0      55       7      30
BII-09        10       1       9       5      50
BI-009        10       3      26      12      31
BII-25        10       5       6      11      52
BII-14        10       5       6       9       7
BII-11        10       5      31      13      14
BII-03        10       6      47      13      21
BI-010        10       7      31      14      19
BII-14        10       9       7      15      46
BII-02        10       9      16      10      10
BII-07        10      10       4      16      49
BII-05        10      10      16      19      33
BII-30        10      10      34      17      13
BII-10        10      11      19      20      34
BI-011        10      11      38      17      58
BI-009        10      11      57      12      31
BI-010        10      13       6      18      23
BII-08        10      13      14      22      28
BII-13        10      15      41      20      49
BII-12        10      16       3      22      40
BII-07        10      16      49      20      40
BI-008        10      16      52      22      34
BII-28        10      16      52      23      21
BII-26        10      18      35       1       6
BII-09        10      19       1       1       5
BII-10        10      19      58      20      43
BII-27        10      20      28       3       7
BII-01        10      20      38       3      12
BII-13        10      20      58       2      27
BII-04        10      22      10       7      21
BI-008        10      22      34       3      50
BII-06        10      23      20       5      59
BII-02        11       0      34       9      12
BII-29        11       0      51       7      26
```

BII-09	11	1	5	5	46
BI-009	11	3	22	11	53
BII-25	11	5	2	11	48
BII-14	11	5	2	9	2
BII-11	11	5	27	13	9
BII-03	11	6	43	13	17
BI-010	11	7	27	14	15
BII-14	11	9	2	15	42
BII-02	11	9	12	10	6
BII-07	11	10	0	16	45
BII-05	11	10	12	19	29
BII-30	11	10	30	17	9
BII-10	11	11	15	19	54
BI-011	11	11	34	17	54
BI-009	11	11	53	12	27
BI-010	11	13	2	18	19
BII-08	11	13	10	22	24
BII-13	11	15	36	20	45
BII-12	11	15	59	22	36
BII-07	11	16	45	20	36
BI-008	11	16	48	22	30
BII-28	11	16	48	23	17
BII-26	11	18	31	1	2
BII-09	11	18	57	1	0
BII-10	11	19	54	20	39
BII-27	11	20	24	3	3
BII-01	11	20	34	3	8
BII-13	11	20	54	2	23
BII-04	11	22	6	7	17
BI-008	11	22	30	3	46
BII-06	11	23	16	5	55
BII-02	12	0	30	9	7
BII-29	12	0	47	7	22
BII-09	12	1	0	5	41
BI-009	12	3	18	11	49
BII-25	12	4	58	11	44
BII-14	12	4	58	8	58
BII-11	12	5	23	13	5
BII-03	12	6	40	13	13
BI-010	12	7	23	14	11
BII-14	12	8	58	15	37
BII-02	12	9	7	10	2
BII-07	12	9	56	16	41
BII-05	12	10	8	19	25
BII-30	12	10	25	17	5
BII-10	12	11	11	19	50
BI-011	12	11	30	17	50
BI-009	12	11	49	12	23
BI-010	12	12	58	18	15
BII-08	12	13	6	22	20
BII-13	12	15	32	20	40

BII-12	12	15	55	22	32
BII-07	12	16	41	20	32
BI-008	12	16	44	22	26
BII-28	12	16	44	23	13
BII-26	12	18	27	0	58
BII-09	12	18	53	0	56
BII-10	12	19	50	20	35
BII-27	12	20	20	2	59
BII-01	12	20	30	3	4
BII-13	12	20	50	2	19
BII-04	12	22	2	7	13
BI-008	12	22	26	3	42
BII-06	12	23	12	5	51
BII-02	13	0	26	9	3
BII-29	13	0	43	7	18
BII-09	13	0	56	5	37
BI-009	13	3	14	11	45
BII-25	13	4	53	11	40
BII-14	13	4	54	8	54
BII-11	13	5	19	13	1
BII-03	13	6	36	13	9
BI-010	13	7	19	14	6
BII-14	13	8	54	15	33
BII-02	13	9	3	9	58
BII-07	13	9	52	16	37
BII-05	13	10	4	19	21
BII-30	13	10	21	17	1
BII-10	13	11	7	19	46
BI-011	13	11	26	17	46
BI-009	13	11	45	12	19
BI-010	13	12	53	18	11
BII-08	13	13	2	22	16
BII-13	13	15	28	20	36
BII-12	13	15	51	22	28
BII-07	13	16	37	20	27
BI-008	13	16	40	22	22
BII-28	[illegible]	16	40	23	9
BII-26	[illegible]	18	22	0	54
BII-09	13	18	49	0	52
BII-10	13	19	46	20	31
BII-27	13	20	16	2	55
BII-01	13	20	26	3	0
BII-13	13	20	46	2	15
BII-04	13	21	58	7	9
BI-008	13	22	22	[illegible]	38
BII-06	13	23	8	[illegible]	47
BII-02	14	0	22	8	59
BII-29	14	0	39	7	13
BII-09	14	0	52	5	33
BI-009	14	3	10	11	40
BII-25	14	4	49	11	36

BII-14	14	4	50	8	50
BII-11	14	5	14	12	56
BII-03	14	6	32	13	5
BI-010	14	7	15	14	2
BII-14	14	8	50	15	29
BII-02	14	8	59	9	53
BII-07	14	9	48	16	33
BII-05	14	9	59	19	17
BII-30	14	10	17	16	57
BII-10	14	11	3	19	42
BI-011	14	11	22	17	42
BI-009	14	11	40	12	15
BI-010	14	12	49	18	7
BII-08	14	12	57	22	12
BII-13	14	15	24	20	32
BII-12	14	15	46	22	23
BII-07	14	16	33	20	23
BI-008	14	16	36	22	18
BII-28	14	16	36	23	5
BII-26	14	18	18	0	50
BII-09	14	18	45	0	48
BII-10	14	19	42	20	27
BII-27	14	20	11	2	50
BII-01	14	20	22	2	56
BII-13	14	20	42	2	11
BII-04	14	21	53	7	5
BI-008	14	22	18	3	34
BII-06	14	23	4	5	42
END 0 0 0 0 0					
CAPE 2					
BII-04	9	18	17	1	0
BII-09	9	19	58	1	19
BII-02	9	20	30	3	50
BI-009	9	22	41	6	49
BII-06	9	22	57	5	43
BII-01	9	23	0	3	1
BII-25	9	23	3	3	42
BII-11	10	0	49	8	52
BI-010	10	1	6	4	49
BII-03	10	2	24	9	18
BII-29	10	2	56	7	10
BII-07	10	3	56	7	54
BII-14	10	5	7	11	22
BI-011	10	5	39	11	3
BII-30	10	6	14	13	8
BII-05	10	6	26	13	6
BII-10	10	6	49	14	45
BII-25	10	6	57	11	24
BI-010	10	8	49	13	38
BII-08	10	8	59	16	5
BII-13	10	9	42	16	13

BI-008	10	10	38	17	44
BII-28	10	10	43	15	45
BI-011	10	11	3	18	13
BII-07	10	11	19	16	18
BII-12	10	11	23	18	59
BII-09	10	12	56	16	10
BII-26	10	14	11	21	1
BII-01	10	14	35	19	33
BII-27	10	15	42	23	32
BII-13	10	16	13	21	58
BII-06	10	17	22	22	53
BI-008	10	17	44	23	14
BII-04	10	18	13	0	56
BII-29	10	18	43	23	33
BII-28	10	19	8	23	6
BII-09	10	19	53	1	15
BII-02	10	20	26	3	46
BI-009	10	22	37	6	45
BII-06	10	22	53	5	39
BII-01	10	22	56	2	57
BII-25	10	22	59	3	38
BII-11	11	0	44	8	48
BI-010	11	1	1	4	45
BII-03	11	2	20	9	14
BII-29	11	2	52	7	6
BII-07	11	3	52	7	50
BII-14	11	5	3	11	18
BI-011	11	5	35	10	59
BII-30	11	6	10	13	4
BII-05	11	6	22	13	2
BII-10	11	6	45	14	41
BII-25	11	6	53	11	20
BI-010	11	8	45	13	33
BII-08	11	8	55	16	1
BII-13	11	9	38	16	9
BI-008	11	10	34	13	4
BII-28	11	10	39	15	41
BI-011	11	10	59	18	9
BII-07	11	11	15	16	14
BII-12	11	11	19	18	55
BII-09	11	12	52	16	6
BII-26	11	14	7	20	57
BII-01	11	14	31	19	28
BII-27	11	15	38	23	28
BII-13	11	16	9	21	54
BII-06	11	17	18	22	49
BI-008	11	17	40	23	10
BII-04	11	18	9	0	52
BII-29	11	18	39	23	29
BII-28	11	19	4	23	2
BII-09	11	19	49	1	11

BII-02	11	20	22	3	41
BI-009	11	22	32	6	41
BII-06	11	22	49	5	35
BII-01	11	22	52	2	53
BII-25	11	22	55	3	34
BII-11	12	0	40	8	44
BI-010	12	0	57	4	41
BII-03	12	2	16	9	10
BII-29	12	2	48	7	2
BII-07	12	3	48	7	46
BII-14	12	4	59	11	14
BI-011	12	5	31	10	55
BII-30	12	6	6	13	0
BII-05	12	6	18	12	58
BII-10	12	6	40	14	37
BII-25	12	6	49	11	15
BI-010	12	8	41	13	29
BII-08	12	8	50	15	57
BII-13	12	9	34	16	5
BI-008	12	10	30	13	0
BII-28	12	10	38	15	37
BI-011	12	10	55	18	5
BII-07	12	11	11	16	9
BII-12	12	11	15	18	51
BII-09	12	12	48	16	2
BII-26	12	14	3	20	53
BII-01	12	14	27	19	24
BII-27	12	15	33	23	24
BII-13	12	16	5	21	49
BII-06	12	17	14	22	45
BI-008	12	17	36	23	6
BII-04	12	18	5	0	48
BII-29	12	18	35	23	25
BII-28	12	19	0	22	58
BII-09	12	19	45	1	7
BII-02	12	20	17	3	37
BI-009	12	22	28	6	37
BII-06	12	22	45	5	31
BII-01	12	22	48	2	49
BII-25	12	22	51	3	30
BII-11	13	0	36	8	40
BI-010	13	0	53	4	37
BII-03	13	2	12	9	6
BII-29	13	2	44	6	58
BII-07	13	3	44	7	42
BII-14	13	4	55	11	10
BI-011	13	5	27	10	51
BII-30	13	6	2	12	55
BII-05	13	6	14	12	54
BII-10	13	6	36	14	33
BII-25	13	6	45	11	11

BI-010	13	8	37	13	25
BII-08	13	8	46	15	53
BII-13	13	9	30	16	1
BI-008	13	10	26	12	56
BII-28	13	10	31	15	32
BI-011	13	10	51	18	1
BII-07	13	11	7	16	5
BII-12	13	11	10	18	47
BII-09	13	12	44	15	58
BI'-26	13	13	58	20	49
BII-01	13	14	23	19	20
BII-27	13	15	29	23	20
BII-13	13	16	1	21	45
BII-06	13	17	10	22	41
BI-008	13	17	31	23	2
BII-04	13	18	1	0	43
BII-29	13	18	30	23	21
BII-28	13	18	56	22	54
BII··09	13	19	41	1	3
BII-02	13	20	13	3	33
BI-009	13	22	24	6	32
BII-06	13	22	41	5	27
BII-u1	13	22	44	2	45
BII-25	13	22	46	3	26
BII-11	14	0	32	8	36
BI-010	14	0	49	4	33
BII-03	14	2	8	9	3
BII-29	14	2	40	6	54
BII-07	14	3	40	7	38
BII-14	14	4	50	11	5
BI-011	14	5	23	10	47
BII-30	14	5	58	12	51
BII-05	14	6	10	12	50
BII-10	14	6	32	14	29
BII-25	14	6	41	11	7
BI-010	14	8	33	13	21
BII-08	14	8	42	15	49
BII-13	14	9	26	15	57
BI-008	14	10	22	12	52
BII-28	14	10	27	15	28
BI-011	14	10	47	17	57
BII-07	14	11	3	16	1
BII-12	14	11	6	18	42
BII-09	14	12	39	15	53
BII-26	14	13	54	20	44
BII-01	14	14	19	19	16
BII-27	14	15	25	23	15
BII-13	14	15	57	21	41
BII-06	14	17	6	22	37
BI-008	14	17	27	22	58
BII-04	14	17	56	0	39

```
BII-29          14        18        26        23        17
BII-28          14        18        52        22        50
BII-09          14        19        37         0        59
BII-02          14        20         9         3        29
BI-009          14        22        20         6        28
BII-06          14        22        37         5        23
BII-01          14        22        40         2        41
BII-25          14        22        42         3        21
END 0 0 0 0 0
DIEG 3
BII-07           9        16        38         1        50
BI-011           9        18        54         0        33
BII-05           9        19        36         2         7
BII-12           9        20        58         3        49
BII-26           9        21        58         6        26
BII-13           9        22        17         7        23
BII-08           9        23        17         4        55
BII-10           9        23        18         3        14
BII-28           9        23        29         6        32
BI-008           9        23        34         5         0
BII-07          10         1        22         1        50
BII-27          10         1        29         8        13
BII-09          10         1        34        10        47
BII-01          10         3        21        10        23
BII-04          10         3        22         7        16
BI-008          10         4        24         9        16
BII-02          10         5        48         6        34
BII-06          10         6         7        12        50
BII-13          10         6        48         7        54
BII-04          10         7        14        13        56
BII-29          10         7        28        14        48
BI-009          10         8        44        14        29
BII-09          10        10        11        11         7
BII-03          10        10        13        18        37
BII-11          10        10        43        17        16
BII-02          10        11         4        16        32
BII-25          10        11        42        19        37
BII-14          10        11        47        21         4
BI-010          10        13        55        23         1
BII-30          10        14         4        22        26
BI-009          10        14        29        19        11
BII-05          10        15        26        19        32
BII-07          10        16        34         1        18
BII-10          10        16        34        23        14
BII-08          10        18         9        23        12
BI-011          10        18        50         0        29
BII-05          10        19        32         2         2
BII-12          10        20        54         3        45
BII-26          10        21        54         6        22
BII-13          10        22        12         6        43
BI-010          10        22        25        23        23
```

BII-08	10	23	12	4	51
BII-10	10	23	14	3	10
BII-28	10	23	25	6	28
BI-008	10	23	30	4	38
BII-07	11	1	18	1	46
BII-27	11	1	25	8	9
BII-09	11	1	30	10	7
BII-01	11	3	17	10	19
BII-04	11	3	18	7	12
BI-008	11	4	20	9	12
BII-02	11	5	44	10	54
BII-06	11	6	3	12	46
BII-13	11	6	43	7	50
BII-04	11	7	10	13	52
BII-29	11	7	24	14	44
BI-009	11	8	40	14	25
BII-09	11	10	7	11	3
BII-03	11	10	9	18	33
BII-11	11	10	39	17	12
BII-02	11	11	0	16	28
BII-25	11	11	37	19	32
BII-14	11	11	43	21	0
BI-010	11	13	51	22	21
BII-30	11	14	0	22	22
BI-009	11	14	25	19	6
BII-05	11	15	22	19	28
BII-07	11	16	30	1	14
BII-10	11	16	30	23	10
BII-08	11	18	5	23	8
BI-011	11	18	46	0	24
BII-05	11	19	28	1	58
BII-12	11	20	50	3	41
BII-26	11	21	49	6	17
BII-13	11	22	8	6	39
BI-010	11	22	21	23	19
BII-08	11	23	8	4	47
BII-10	11	23	10	3	6
BII-28	11	23	21	6	24
BI-008	11	23	26	4	34
BII-07	12	1	14	1	42
BII-27	12	1	21	8	5
BII-09	12	1	26	10	3
BII-01	12	3	12	10	15
BII-04	12	3	13	7	8
BI-008	12	4	16	9	8
BII-02	12	5	40	10	50
BII-06	12	5	59	12	42
BII-13	12	6	39	7	46
BII-04	12	7	6	13	47
BII-29	12	7	20	14	40
BI-009	12	8	35	14	21

BII-09	12	10	3	10	58
BII-03	12	10	6	18	29
BII-11	12	10	35	17	7
BII-02	12	10	56	16	24
BII-25	12	11	33	19	28
BII-14	12	11	39	20	55
BI-010	12	13	47	22	17
BII-30	12	13	56	22	18
BI-009	12	14	21	19	2
BII-05	12	15	18	19	24
BII-07	12	16	25	1	9
BII-10	12	16	26	23	6
BII-08	12	18	1	23	4
BI-011	12	18	42	0	20
BII-05	12	19	24	1	54
BII-12	12	20	45	3	37
BII-26	12	21	45	6	13
BII-13	12	22	4	6	35
BI-010	12	22	17	23	15
BII-08	12	23	4	4	43
BII-10	12	23	6	?	2
BII-28	12	23	17	6	20
BI-008	12	23	22	4	30
BII-07	13	1	9	1	38
BII-27	13	1	17	8	1
BII-09	13	1	22	9	59
BII-01	13	3	8	10	11
BII-04	13	3	9	7	4
BI-008	13	4	12	9	4
BII-02	13	5	35	10	46
BII-06	13	5	55	12	38
BII-13	13	6	35	7	41
BII-04	13	7	2	13	43
BII-29	13	7	16	14	35
BI-009	13	8	31	14	17
BII-09	13	9	59	10	54
BII-03	13	10	2	18	25
BII-11	13	10	31	17	3
BII-02	13	10	52	16	19
BII-25	13	11	29	19	24
BII-14	13	11	35	20	51
BI-010	13	13	43	22	13
BII-30	13	13	52	22	14
BI-009	13	14	17	18	58
BII-05	13	15	14	19	20
BII-07	13	16	21	1	5
BII-10	13	16	22	23	2
BII-08	13	17	57	23	0
BI-011	13	18	38	0	16
BII-05	13	19	20	1	50
BII-12	13	20	41	3	32

BII-26	13	21	41	6	9
BII-13	13	22	0	6	31
BI-010	13	22	13	23	11
BII-08	13	23	0	4	39
BII-10	13	23	2	2	58
BII-28	13	23	13	6	16
BI-008	13	23	18	4	25
BII-07	14	1	5	1	33
BII-27	14	1	13	7	57
BII-09	14	1	18	9	55
BII-01	14	3	4	10	7
BII-04	14	3	5	7	0
BI-008	14	4	8	9	0
BII-02	14	5	31	10	42
BII-06	14	5	51	12	34
BII-13	14	6	31	7	37
BII-04	14	6	57	13	39
BII-29	14	7	12	14	31
BI-009	14	8	27	14	12
BII-09	14	9	55	10	50
BII-03	14	9	58	18	21
BII-11	14	10	26	16	59
BII-02	14	10	48	16	15
BII-25	14	11	25	19	20
BII-14	14	11	31	20	47
BI-010	14	13	3[illegible]	22	9
BII-30	14	13	48	22	10
BI-009	14	14	12	18	54
BII-05	14	15	10	19	16
BII-07	14	16	17	1	1
BII-10	14	16	18	22	58
BII-08	14	17	53	22	56
BI-011	14	18	34	0	12
BII-05	14	19	16	1	46
BII-12	14	20	37	3	28
BII-26	14	21	36	6	5
BII-13	14	21	56	6	27
BI-010	14	22	9	23	7
BII-08	14	22	56	4	35
BII-10	14	22	58	2	54
BII-28	14	23	9	6	12
BI-008	14	23	14	4	21
END 0 0 0 0 0					
KWAJ 4					
BII-03	9	17	13	0	42
BII-25	9	17	53	2	29
BII-14	9	17	59	3	13
BI-010	9	20	13	5	8
BI-009	9	20	25	1	40
BII-30	9	21	7	4	34
BII-10	9	22	46	4	55

BII-07	9	22	49	8	5
BII-08	10	0	23	4	57
BI-011	10	0	56	6	47
BII-05	10	1	21	8	16
BII-12	10	3	14	9	48
BII-13	10	4	27	13	44
BI-010	10	4	32	5	46
BII-10	10	4	55	9	33
BII-08	10	4	57	11	11
BII-26	10	5	2	12	27
BI-008	10	5	37	10	59
BII-28	10	5	41	13	30
BII-07	10	7	31	8	5
BII-27	10	7	43	14	8
BII-09	10	7	46	17	1
BII-01	10	9	28	17	16
BII-04	10	9	36	13	0
BI-008	10	9	56	15	17
BII-02	10	12	2	16	49
BII-06	10	12	20	19	8
BII-04	10	13	0	20	6
BII-29	10	13	38	22	3
BI-009	10	14	47	22	45
BII-09	10	16	25	17	12
BII-02	10	16	49	22	50
BII-11	10	16	49	23	17
BII-03	10	17	10	0	38
BII-25	10	17	49	2	25
BII-14	10	17	55	3	9
BI-010	10	20	9	4	28
BI-009	10	20	20	1	36
BII-30	10	21	3	4	30
BII-05	10	21	35	1	17
BII-10	10	22	42	4	51
BII-07	10	22	44	7	27
BII-08	11	0	19	4	53
BI-011	11	0	52	6	43
BII-05	11	1	17	8	12
BII-12	11	3	10	9	44
BII-13	11	4	23	13	4
BI-010	11	4	28	5	42
BII-10	11	4	51	9	29
BII-08	11	4	53	11	7
BII-26	11	4	58	12	23
BI-008	11	5	33	9	52
BII-28	11	5	37	13	26
BII-07	11	7	27	8	1
BII-27	11	7	39	14	4
BII-09	11	7	42	16	21
BII-01	11	9	24	17	12
BII-04	11	9	31	12	56

BI-008	11	9	52	15	13
BII-02	11	11	57	16	45
BII-06	11	12	16	19	4
BII-04	11	12	56	20	2
BII-29	11	13	33	21	59
BI-009	11	14	43	22	41
BII-09	11	16	21	17	8
BII-02	11	16	45	22	46
BII-11	11	16	45	23	13
BII-03	11	17	6	0	34
BII-25	11	17	45	2	21
BII-14	11	17	51	3	5
BI-010	11	20	5	4	24
BI-009	11	20	16	1	31
BII-30	11	20	59	4	26
BII-05	11	21	31	1	13
BII-10	11	22	38	4	47
BII-07	11	22	40	7	23
BII-08	12	0	14	4	49
BI-011	12	0	48	6	39
BII-05	12	1	13	8	8
BII-12	12	3	6	9	40
BII-13	12	4	18	13	0
BI-010	12	4	24	5	38
BII-10	12	4	47	9	25
BII-08	12	4	49	11	3
BII-26	12	4	54	12	19
BI-008	12	5	29	9	48
BII-28	12	5	33	13	22
BII-07	12	7	23	7	56
BII-27	12	7	35	14	0
BII-09	12	7	38	16	17
BII-01	12	9	20	17	9
BII-04	12	9	27	12	51
BI-008	12	9	48	15	9
BII-02	12	11	53	16	40
BII-06	12	12	12	19	0
BII-04	12	12	51	19	58
BII-29	12	13	29	21	55
BI-009	12	14	39	22	37
BII-09	12	16	17	17	4
BII-02	12	16	40	22	42
BII-11	12	16	41	23	8
BII-03	12	17	2	0	30
BII-25	12	17	41	2	17
BII-14	12	17	47	3	1
BI-010	12	20	1	4	20
BI-009	12	20	12	1	27
BII-30	12	20	55	4	21
BII-05	12	21	27	1	9
BII-10	12	22	34	4	43

BII-07	12	22	36	7	19
BII-08	13	0	10	4	45
BI-011	13	0	44	6	35
BII-05	13	1	9	8	4
BII-12	13	3	1	9	36
BII-13	13	4	14	12	56
BI-010	13	4	20	5	34
BII-10	13	4	43	9	21
BII-08	13	4	45	10	59
BII-26	13	4	49	12	15
BI-008	13	5	25	9	44
BII-28	13	5	28	13	18
BII-07	13	7	19	7	52
BII-27	13	7	31	13	56
BII-09	13	7	34	16	13
BII-01	13	9	16	17	5
BII-04	13	9	23	12	47
BI-008	13	9	44	15	5
BII-02	13	11	49	16	36
BII-06	13	12	8	18	56
BII-04	13	12	47	19	53
BII-29	13	13	25	21	51
BI-009	13	14	35	22	33
BII-09	13	16	13	17	0
BII-02	13	16	36	22	38
BII-11	13	16	37	23	4
BII-03	13	16	58	0	26
BII-25	13	17	36	2	13
BII-14	13	17	43	2	56
BI-010	13	19	57	4	16
BI-009	13	20	8	1	23
BII-30	13	20	50	4	17
BII-05	13	21	23	1	5
BII-10	13	22	30	4	39
BII-07	13	22	32	7	15
BII-08	14	0	6	4	41
BI-011	14	0	40	6	31
BII-05	14	1	5	8	0
BII-12	14	2	57	9	32
BII-13	14	4	10	12	52
BI-010	14	4	16	5	30
BII-10	14	4	39	9	17
BII-08	14	4	41	10	55
BII-26	14	4	45	12	11
BI-008	14	5	21	9	40
BII-28	14	5	24	13	14
BII-07	14	7	15	7	48
BII-27	14	7	27	13	52
BII-09	14	7	30	16	9
BII-01	14	9	12	17	1
BII-04	14	9	19	12	43

```
BI-008      14     9     40     15      1
BII-02      14    11     45     16     32
BII-06      14    12      4     18     52
BII-04      14    12     43     19     49
BII-29      14    13     21     21     46
BI-009      14    14     30     22     29
BII-09      14    16      9     16     55
BII-02      14    16     32     22     34
BII-11      14    16     33     23      0
BII-03      14    16     54      0     22
BII-25      14    17     32      2      9
BII-14      14    17     38      2     52
BI-010      14    19     53      4     12
BI-009      14    20      4      1     19
BII-30      14    20     46      4     13
BII-05      14    21     19      1      1
BII-10      14    22     26      4     35
BII-07      14    22     28      7     10
END 0 0 0 0 0
PIKE 5
BII-04       9    17      4      0     16
BII-02       9    19     37      3      1
BII-09       9    21      8      1      7
BII-14       9    21     43      1     17
BII-25       9    22      4      3     34
BI-009       9    22     35      5     39
BII-30       9    23     10      1     25
BI-010      10     0     16      5     11
BII-01      10     0     41      2     50
BII-11      10     0     47      7     38
BII-03      10     2     49      8     30
BII-07      10     2     59      8      6
BI-011      10     4     18     10     39
BII-29      10     4     33      6     59
BII-05      10     5     12     12     22
BII-14      10     5     48     10     55
BII-10      10     6     25     13     46
BII-30      10     6     36     12     18
BII-26      10     7      3      9     25
BII-08      10     7     59     15     16
BII-25      10     8     33     11     13
BII-13      10     8     35     12     51
BII-28      10     9     41     15     30
BI-008      10     9     46     13     57
BI-010      10     9     52     13     35
BII-12      10    11     34     17     48
BII-09      10    11     56     16     40
BII-07      10    12     41     16      8
BII-01      10    13     34     19     17
BII-26      10    14     36     20     14
BII-27      10    15     50     22     19
```

BII-06	10	16	15	22	25
BII-04	10	17	0	0	12
BII-13	10	17	12	21	42
BI-011	10	17	15	18	19
BII-29	10	17	42	23	23
BI-008	10	18	33	23	6
BII-03	10	19	15	21	35
BII-02	10	19	33	2	56
BII-28	10	20	48	22	55
BII-09	10	21	4	1	3
BII-14	10	21	39	1	13
BII-25	10	22	0	3	29
BI-009	10	22	31	5	35
BII-30	10	23	6	1	21
BI-010	11	0	12	5	7
BII-01	11	0	36	2	46
BII-11	11	0	43	7	34
BII-03	11	2	45	8	26
BII-07	11	2	55	8	2
BI-011	11	4	14	10	35
BII-29	11	4	29	6	55
BII-05	11	5	8	12	18
BII-14	11	5	43	10	51
BII-10	11	6	21	13	41
BII-30	11	6	34	12	14
BII-26	11	6	59	9	21
BII-08	11	7	55	15	12
BII-25	11	8	29	11	9
BII-13	11	8	31	12	47
BII-28	11	9	37	15	26
BI-008	11	9	42	13	53
BI-010	11	9	48	13	31
BII-12	11	11	30	17	44
BII-09	11	11	52	16	36
BII-07	11	12	37	16	3
BII-01	11	13	30	19	13
BII-26	11	14	32	20	9
BII-27	11	15	46	22	14
BII-06	11	16	11	22	21
BII-04	11	16	53	0	8
BII-13	11	17	8	21	38
BI-011	11	17	11	18	15
BII-29	11	17	37	23	19
BI-008	11	18	29	23	2
BII-03	11	19	11	21	31
BII-02	11	19	28	2	52
BII-28	11	20	44	22	51
BII-09	11	21	0	0	59
BII-14	11	21	35	1	9
BII-25	11	21	55	3	25
BI-009	11	22	27	5	31

BII-30	11	23	2	1	17
BI-010	12	0	8	5	2
BII-01	12	0	32	2	42
BII-11	12	0	38	7	30
BII-03	12	2	41	8	22
BII-07	12	2	51	7	57
BI-011	12	4	10	10	31
BII-29	12	4	25	6	51
BII-05	12	5	4	12	14
BII-14	12	5	39	10	46
BII-10	12	6	17	13	37
BII-30	12	6	30	12	10
BII-26	12	6	55	9	17
BII-08	12	7	51	15	8
BII-25	12	8	25	11	5
BII-13	12	8	26	12	43
BII-28	12	9	33	15	22
BI-008	12	9	37	13	49
BI-010	12	9	44	13	27
BII-12	12	11	26	17	40
BII-09	12	11	48	16	32
BII-07	12	12	33	15	59
BII-01	12	13	26	19	9
BII-26	12	14	27	20	5
BII-27	12	15	41	22	10
BII-06	12	16	7	22	17
BII-04	12	16	51	0	4
BII-13	12	17	4	21	34
BI-011	12	17	7	18	11
BII-29	12	17	33	23	15
BI-008	12	18	25	22	58
BII-03	12	19	7	21	27
BII-02	12	19	24	2	48
BII-28	12	20	40	22	46
BII-09	12	20	56	0	55
BII-14	12	21	31	1	6
BII-25	12	21	51	3	21
BI-009	12	22	23	5	27
BII-30	12	22	58	1	13
BI-010	13	0	4	4	58
BII-01	13	0	28	2	38
BII-11	13	0	34	7	26
BII-03	13	2	37	8	18
BII-07	13	2	47	7	53
BI-011	13	4	6	10	27
BII-29	13	4	21	6	47
BII-05	[illegible]	5	0	12	10
BII-14	13	5	35	10	42
BII-10	13	6	12	13	33
BII-30	13	6	26	12	6
BII-26	13	6	50	9	13

BII-08	13	7	47	15	4
BII-25	13	8	21	11	1
BII-13	13	8	22	12	39
BII-28	13	9	20	15	18
BI-008	13	9	33	13	45
BI-010	13	9	40	13	23
BII-12	13	11	22	17	36
BII-09	13	11	44	16	28
BII-07	13	12	29	15	55
BII-01	13	13	22	19	5
BII-26	13	14	23	20	1
BII-27	13	15	37	22	6
BII-06	13	16	3	22	13
BII-04	13	16	47	0	0
BII-13	13	17	0	21	29
BI-011	13	17	3	18	7
BII-29	13	17	29	23	11
BI-008	13	18	21	22	54
BII-03	13	19	3	21	23
BII-02	13	19	20	2	44
BII-28	13	20	36	22	42
BII-09	13	20	52	0	51
BII-14	13	21	27	1	1
BII-25	13	21	47	3	17
BI-009	13	22	19	5	23
BII-30	13	22	54	1	9
BI-010	14	0	0	4	54
BII-01	14	0	24	2	34
BII-11	14	0	30	7	21
BII-03	14	2	33	8	14
BII-07	14	2	42	7	49
BI-011	14	4	2	10	23
BII-29	14	4	17	6	43
BII-05	14	4	56	12	6
BII-14	14	5	31	10	38
BII-10	14	6	8	13	29
BII-30	14	6	21	12	2
BII-26	14	6	46	9	9
BII-08	14	7	43	15	0
BII-25	14	8	17	10	56
BII-13	14	8	18	12	35
BII-28	14	9	25	15	14
BI-008	14	9	29	13	41
BI-010	14	9	36	13	19
BII-12	14	11	18	17	31
BII-09	14	11	40	16	24
BII-07	14	12	24	15	51
BII-01	14	13	18	19	1
BII-26	14	14	19	19	57
BII-27	14	15	33	22	2
BII-06	14	15	59	22	9

```
BII-04          14        16        43        23        56
BII-13          14        16        56        21        25
BI-011          14        16        59        18         2
BII-29          14        17        25        23         7
BI-008          14        18        17        22        50
BII-03          14        18        59        21        19
BII-02          14        19        16         2        40
BII-28          14        20        32        22        38
BII-09          14        20        48         0        47
BII-14          14        21        23         0        57
BII-25          14        21        43         3        13
BI-009          14        22        15         5        19
BII-30          14        22        50         1         5
BI-010          14        23        56         4         0
END 0 0 0 0 0
```

B.5 Simulation Time and Parameters.

This short data file is pointed to by the value in the TIME.DAT variable and contains the simulation start and end times and three simulation parameters.

```
100 00 00
102 00 00
3
3
10
```

B.6 Ground Antenna Outage Data.

This data file is pointed to by the value in the OUTAGE.DAT variable and contains textual labels, then a list of GA index numbers and outage time data. The simulation will not have the GAs listed available for use during the periods described. A text description of the reason for the outage is also provided

```
"ASCN out for entire simulation."
"DIEG out for entire simulation."
2
1  144000  2880  OUTAGE
3  144000  2880  OUTAGE
```

B.7 *Preliminary Activity Data.*

To schedule activities immediately at the start of the simulation, the MCS Simulation program requires information on the activities that occured just prior to the start time. The file pointed to by the variable ACT.DAT contains this data. The file below contains the activities for all 24 test SVs.

```
24
93
BI-008 1 ADDKEYS  143815  10 1 1       4320
BI-008 1 NAV      143232   5 1 1       1440
BI-008 1 SOH      143805  10 1 1        480
BI-009 1 ADDKEYS  142703  10 1 1       4320
BI-009 1 NAV      142713   5 1 1       1440
BI-009 1 SOH      143758  10 1 1        480
BI-010 1 ADDKEYS  143783  10 1 1       4320
BI-010 1 NAV      142800   5 1 1       1440
BI-010 1 SOH      143773  10 1 1        480
BI-011 1 ADDKEYS  143408  10 1 1       4320
BI-011 1 NAV      142975   5 1 1       1440
BI-011 1 SOH      143922  10 1 1        480
BII-01 2 GBDDUMP  143508  10 1 1        720
BII-01 2 NAV      143518  15 1 1       1440
BII-01 2 NAVBUFF  142808   5 1 1       1440
BII-01 2 SOH      143498  10 1 1        720
BII-02 2 GBDDUMP  143863  10 1 1        720
BII-02 2 NAV      143873  15 1 1       1440
BII-02 2 NAVBUFF  143302   5 1 1       1440
BII-02 2 SOH      143853  10 1 1        720
BII-03 2 GBDDUMP  143841  10 1 1        720
BII-03 2 NAV      143851  15 1 1       1440
BII-03 2 NAVBUFF  143090   5 1 1       1440
BII-03 2 SOH      143831  10 1 1        720
BII-04 2 GBDDUMP  143770  10 1 1        720
BII-04 2 NAV      143780  15 1 1       1440
BII-04 2 NAVBUFF  142925   5 1 1       1440
BII-04 2 SOH      143760  10 1 1        720
BII-05 2 GBDDUMP  143900  10 1 1        720
BII-05 2 NAV      143075  15 1 1       1440
BII-05 2 NAVBUFF  143910   5 1 1       1440
BII-05 2 SOH      143890  10 1 1        720
BII-06 2 GBDDUMP  143637  10 1 1        720
BII-06 2 NAV      143647  15 1 1       1440
BII-06 2 NAVBUFF  142891   5 1 1       1440
BII-06 2 SOH      143627  10 1 1        720
BII-07 2 GBDDUMP  143576  10 1 1        720
```

BII-07	2	NAV	142918	15	1	1	1440
BII-07	2	NAVBUFF	143586	5	1	1	1440
BII-07	2	SOH	143566	10	1	1	720
BII-08	2	GBDDUMP	143837	10	1	1	720
BII-08	2	NAV	143243	15	1	1	1440
BII-08	2	NAVBUFF	143847	5	1	1	1440
BII-08	2	SOH	143827	10	1	1	720
BII-09	2	GBDDUMP	143477	10	1	1	720
BII-09	2	NAV	143487	15	1	1	1440
BII-09	2	NAVBUFF	142835	5	1	1	1440
BII-09	2	SOH	143467	10	1	1	720
BII-10	3	GBDDUMP	143898	10	1	1	720
BII-10	3	NAV	143125	20	1	1	1440
BII-10	3	NAVBUFF	143908	5	1	1	1440
BII-10	3	SOLAR	143878	15	1	1	360
BII-10	3	SOH	143888	10	1	1	720
BII-11	3	GBDDUMP	143455	10	1	1	720
BII-11	3	NAV	142831	20	1	1	1440
BII-11	3	NAVBUFF	143465	5	1	1	1440
BII-11	3	SOH	143445	10	1	1	720
BII-12	3	NDUTLM	143299	10	1	1	720
BII-12	3	NAV	143309	20	1	1	1440
BII-12	3	NAVBUFF	142742	5	1	1	1440
BII-12	3	SOH	143289	10	1	1	720
BII-13	3	GBDDUMP	143980	10	1	1	720
BII-13	3	NAV	143260	20	1	1	1440
BII-13	3	NAVBUFF	143990	5	1	1	1440
BII-13	3	SOH	143970	10	1	1	720
BII-14	3	GBDDUMP	143400	10	1	1	720
BII-14	3	NAV	142700	20	1	1	1440
BII-14	3	NAVBUFF	143410	5	1	1	1440
BII-14	3	SOH	143390	10	1	1	720
BII-25	3	GBDDUMP	143650	10	1	1	720
BII-25	3	NAV	143660	20	1	1	1440
BII-25	3	NAVBUFF	142960	5	1	1	1440
BII-25	3	SOH	143640	10	1	1	720
BII-26	3	GBDDUMP	143920	10	1	1	720
BII-26	3	NAV	143150	20	1	1	1440
BII-26	3	NAVBUFF	142930	5	1	1	1440
BII-26	3	SOH	143910	10	1	1	720
BII-27	3	GBDDUMP	143350	10	1	1	720
BII-27	3	NAV	142640	20	1	1	1440
BII-27	3	NAVBUFF	143360	5	1	1	1440
BII-27	3	SOH	143340	10	1	1	720
BII-28	3	GBDDUMP	143910	10	1	1	720
BII-28	3	NAV	143920	20	1	1	1440
BII-28	3	NAVBUFF	143310	5	1	1	1440
BII-28	3	SOH	143900	10	1	1	720
BII-29	3	GBDDUMP	143815	10	1	1	720
BII-29	3	NAV	143825	20	1	1	1440
BII-29	3	NAVBUFF	143120	5	1	1	1440

```
BII-29 3 SOH      143805  10 1 1        720
BII-30 3 GBDDUMP  143440  10 1 1        720
BII-30 3 WAV      142775  20 1 1       1440
BII-30 3 WAVBUFF  143450   5 1 1       1440
BII-30 3 SOH      143430  10 1 1        720
```

Appendix C. *MCS Simulation Validation Data/Results*

C.1 Validation Activity Data.

Following is the data used in the validation of the MCS Simulation. These data describe 57 actual activities that were scheduled at the MCS on J100 (10 Apr) 1992. The data were used to compare the scheduling performance of the simulation to the schedule created at the MCS. The format is identical to the format of the data in ACT.FILE, except that the GA at which the activity was historically scheduled is appended.

```
57
BI-008 1 NAV       144660   5 1 1       1440 CAPE
BI-008 1 SOH       144200  10 1 1        480 DIEG
BI-009 1 NAV       144145   5 1 1       1440 CAPE
BI-009 1 SOH       144135  10 1 1        480 CAPE
BI-010 1 NAV       144235   5 1 1       1440 KWAJ
BI-010 1 SOH       144225  10 1 1        480 KWAJ
BI-011 1 NAV       144385   5 1 1       1440 PIKE
BI-011 1 SOH       144375  10 1 1        480 PIKE
BII-01 2 GBDDUMP   144220  10 1 1        720 DIEG
BII-01 2 NAV       144950  15 1 1       1440 KWAJ
BII-01 2 NAVBUFF   144230   5 1 1       1440 DIEG
BII-01 2 SOH       144210  10 1 1        720 DIEG
BII-02 2 GBDDUMP   144730  10 1 1        720 DIEG
BII-02 2 NAV       145310  15 1 1       1440 KWAJ
BII-02 2 NAVBUFF   144740   5 1 1       1440 DIEG
BII-02 2 SOH       144720  10 1 1        720 DIEG
BII-03 2 GBDDUMP   144520  10 1 1        720 CAPE
BII-03 2 NAV       145280  15 1 1       1440 PIKE
BII-03 2 NAVBUFF   144530   5 1 1       1440 CAPE
BII-03 2 SOH       144510  10 1 1        720 CAPE
BII-04 2 GBDDUMP   144355  10 1 1        720 DIEG
BII-04 2 NAV       145115  15 1 1       1440 KWAJ
BII-04 2 NAVBUFF   144365   5 1 1       1440 DIEG
BII-04 2 SOH       144345  10 1 1        720 DIEG
BII-05 2 GBDDUMP   144505  10 1 1        720 PIKE
BII-05 2 NAV       144515  15 1 1       1440 PIKE
BII-05 2 NAVBUFF   145340   5 1 1       1440 KWAJ
BII-05 2 SOH       144495  10 1 1        720 PIKE
BII-06 2 GBDDUMP   144310  10 1 1        720 ASCN
BII-06 2 NAV       145070  15 1 1       1440 KWAJ
BII-06 2 NAVBUFF   144320   5 1 1       1440 ASCN
BII-06 2 SOH       144300  10 1 1        720 ASCN
```

BII-07	2	GBDDUMP	144350	10	1	1	720	PIKE
BII-07	2	NAV	144360	15	1	1	1440	PIKE
BII-07	2	NAVBUFF	145025	5	1	1	1440	DIEG
BII-07	2	SOH	144340	10	1	1	720	PIKE
BII-08	2	GBDDUMP	144655	10	1	1	720	PIKE
BII-08	2	NAV	144665	15	1	1	1440	PIKE
BII-08	2	NAVBUFF	145280	5	1	1	1440	DIEG
BII-08	2	SOH	144645	10	1	1	720	PIKE
BII-09	2	GBDDUMP	144260	10	1	1	720	ASCN
BII-09	2	NAV	144905	15	1	1	1440	KWAJ
BII-09	2	NAVBUFF	144270	5	1	1	1440	ASCN
BII-09	2	SOH	144250	10	1	1	720	ASCN
BII-10	3	GBDDUMP	144550	10	1	1	720	PIKE
BII-10	3	NAV	144560	20	1	1	1440	PIKE
BII-10	3	NAVBUFF	145310	5	1	1	1440	DIEG
BII-10	3	SOLAR	144240	15	1	1	360	KWAJ
BII-10	3	SOH	144540	10	1	1	720	PIKE
BII-11	3	GBDDUMP	144265	10	1	1	720	CAPE
BII-11	3	NAV	144275	20	1	1	1440	CAPE
BII-11	3	NAVBUFF	144905	5	1	1	1440	DIEG
BII-11	3	SOH	144255	10	1	1	720	CAPE
BII-12	3	NDUTLM	144175	10	1	1	720	DIEG
BII-12	3	NAV	144740	20	1	1	1440	CAPE
BII-12	3	NAVBUFF	144185	5	1	1	1440	DIEG
BII-12	3	SOH	144165	10	1	1	720	DIEG

C.2 Validation Results.

There were 14 validation runs made. The object was to "tune" the simulation parameters affecting scheduling to maximize the correlation between a set of MCS-scheduled activities and that same set of activities scheduled by the simulation. The raw results of these runs are below.

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V1 - TBS rank: hP/hL/hI

 9 Prescheduled outages:
   Index     Start   Duration    Cause

     1      144000        90    CPU_OUT
     2      144000        90    CPU_OUT
     3      144000        90    CPU_OUT
     4      144000        90    CPU_OUT
     5      144000        90    CPU_OUT
     1      144120       120    PMI
     3      144420       120    PMI
     4      144600       120    PMI
     2      145080       359    LAUNCH

RESULTS:
              Number     Max        Min         Mean      Std Dev

OFFSET:           92      90          0       1.9565      13.1247
PRIORITY:         92       3          0       0.0652       0.4375
GA UTIL:        1440       3          0       0.7132       0.7864
GA RESV:        1440       5          0       1.5313       1.2396


PRIORITY:
   Value  Number  Percent

     0      90     97.83
     1       0      0.
     2       0      0.
     3       2      2.17
     4       0      0.
     5       0      0.
     6       0      0.
```

```
       7          0        0.
       8          0        0.
       9          0        0.
      10          0        0.
      11          0        0.

GA UTILIZATION/RESERVATION:
   Value  Number          Percent

      0    678/ 227      47.08/ 15.76
      1    530/ 632      36.81/ 43.89
      2    199/ 368      13.82/ 25.56
      3     33/ 106       2.29/  7.36

OFFSET:
   Value  Number  Percent

      0     90     97.83
     10      0      0.
     20      0      0.
     30      0      0.
     40      0      0.
     50      0      0.
     60      0      0.
     70      0      0.
     80      0      0.
     90      2      2.17
    100      0      0.
    110      0      0.
    120      0      0.
    130      0      0.
    140      0      0.
    150      0      0.
    160      0      0.
    170      0      0.
    180      0      0.
    190      0      0.
    200      0      0.
    210      0      0.
    220      0      0.
    230      0      0.
    240      0      0.

 SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V1 - TBS rank: hP/hL/hI
```

```
VALIDATION RESULTS:
                     Number    Max      Min        Mean      Std Dev


START OFFSET:          57     148        0       41.1404    40.9769
GA MATCH:              23


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      36     63.16
    50      15     26.32
   100       6     10.53
   150       0      0.
   200       0      0.
   250       0      0.
   300       0      0.
   350       0      0.
   400       0      0.
   450       0      0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V2 - TBS rank: hP/hL/1SV

VALIDATION RESULTS:
                  Number    Max       Min        Mean      Std Dev


START OFFSET:       57     148         0      41.1404    40.9769
GA MATCH:           23


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      36      63.16
    50      15      26.32
   100       6      10.53
   150       0       0.
   200       0       0.
   250       0       0.
   300       0       0.
   350       0       0.
   400       0       0.
   450       0       0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V3 - TBS rank: hP/1L/hI


VALIDATION RESULTS:
                 Number    Max       Min         Mean      Std Dev


START OFFSET:      57     169         0       41.9825    43.0053
GA MATCH:          25


ACTIVITY START OFFSET:
   Value  Number  Percent

      0     37     64.91
     50     13     22.81
    100      6     10.53
    150      1      1.75
    200      0      0.
    250      0      0.
    300      0      0.
    350      0      0.
    400      0      0.
    450      0      0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V4 - TBS rank: hP/1L/1SV


VALIDATION RESULTS:
                Number    Max       Min         Mean       Std Dev


START OFFSET:     57      168        0        41.9825     43.0053
GA MATCH:         25


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      37      64.91
    50      13      22.81
   100       6      10.53
   150       1       1.75
   200       0       0.
   250       0       0.
   300       0       0.
   350       0       0.
   400       0       0.
   450       0       0.
```

```
SCHEDULING INPUTS
  TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
  16 SVs
   5 GAs
   3 Max contacts
  Actual GA J100 maint, w/ j100 CPU outage
  Pike full availability
  V5 - TBS rank: hP/1SV/1L


VALIDATION RESULTS:
                  Number    Max       Min         Mean      Std Dev


START OFFSET:       57      148         0        40.8947    41.0565
GA MATCH:           25


ACTIVITY START OFFSET:
   Value  Number  Percent

      0     37     64.91
     50     14     24.56
    100      6     10.53
    150      0      0.
    200      0      0
    250      0      0.
    300      0      0.
    350      0      0.
    400      0      0.
    450      0      0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V6 - TBS rank: hP/lSV/hL


VALIDATION RESULTS:
                    Number    Max       Min        Mean      Std Dev


START OFFSET:         57     148         0       40.8947    41.0565
GA MATCH:             25


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      37     64.91
    50      14     24.56
   100       6     10.53
   150       0      0.
   200       0      0.
   250       0      0.
   300       0      0.
   350       0      0.
   400       0      0.
   450       0      0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V7 - TBS rank: hP/hI


VALIDATION RESULTS:
                    Number    Max       Min        Mean      Std Dev


START OFFSET:         57      148        0       40.8246    41.0648
GA MATCH:             25


ACTIVITY START OFFSET:
   Value  Number  Percent

      0     20     35.09
     10      7     12.28
     20      3      5.26
     30      3      5.26
     40      4      7.02
     50      2      3.51
     60      4      7.02
     70      1      1.75
     80      3      5.26
     90      4      7.02
    100      2      3.51
    110      2      3.51
    120      0      0.
    130      0      0.
    140      2      3.51
    150      0      0.
    160      0      0.
    170      0      0.
    180      0      0.
    190      0      0.
    200      0      0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V8 - TBS rank: hP/lI


VALIDATION RESULTS:
                   Number    Max       Min        Mean      Std Dev


START OFFSET:        57     148         0       40.5965    40.8355
GA MATCH:            26


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      20      35.09
    10       7      12.28
    20       3       5.26
    30       3       5.26
    40       4       7.02
    50       2       3.51
    60       5       8.77
    70       0       0.
    80       3       5.26
    90       4       7.02
   100       2       3.51
   110       2       3.51
   120       0       0.
   130       0       0.
   140       2       3.51
   150       0       0.
   160       0       0.
   170       0       0.
   180       0       0.
   190       0       0.
   200       0       0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V9 - TBS rank: hP/hD


VALIDATION RESULTS:
                  Number    Max       Min         Mean      Std Dev


START OFFSET:       57     148         0       40.5965   40.8355
GA MATCH:           26


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      20      35.09
    10       7      12.28
    20       3       5.26
    30       3       5.26
    40       4       7.02
    50       2       3.51
    60       5       8.77
    70       0       0.
    80       3       5.26
    90       4       7.02
   100       2       3.51
   110       2       3.51
   120       0       0.
   130       0       0.
   140       2       3.51
   150       0       0.
   160       0       0.
   170       0       0.
   180       0       0.
   190       0       0.
   200       0       0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V10 - TBS rank: hP/1D


VALIDATION RESULTS:
                  Number    Max       Min        Mean      Std Dev


START OFFSET:       57     148        0       40.8246    41.0648
GA MATCH:           25


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      20     35.09
    10       7     12.28
    20       3      5.26
    30       3      5.26
    40       4      7.02
    50       2      3.51
    60       4      7.02
    70       1      1.75
    80       3      5.26
    90       4      7.02
   100       2      3.51
   110       2      3.51
   120       0      0.
   130       0      0.
   140       2      3.51
   150       0      0.
   160       0      0.
   170       0      0.
   180       0      0.
   190       0      0.
   200       0      0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V11 - TBS rank: hP/1I/hSV


VALIDATION RESULTS:
                      Number    Max       Min        Mean      Std Dev


START OFFSET:          57      148        0       40.6667    40.8276
GA MATCH:              26


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      20      35.09
    10       7      12.28
    20       3       5.26
    30       3       5.26
    40       4       7.02
    50       2       3.51
    60       5       8.77
    70       0       0.
    80       3       5.26
    90       4       7.02
   100       2       3.51
   110       2       3.51
   120       0       0.
   130       0       0.
   140       2       3.51
   150       0       0.
   160       0       0.
   170       0       0.
   180       0       0.
   190       0       0.
   200       0       0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V12 - TBS rank: hP/1I/1SV


VALIDATION RESULTS:
                  Number    Max       Min         Mean      Std Dev


START OFFSET:       57      148         0       40.5965    40.8355
GA MATCH:           26


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      20      35.09
    10       7      12.28
    20       3       5.26
    30       3       5.26
    40       4       7.02
    50       2       3.51
    60       5       8.77
    70       0       0.
    80       3       5.26
    90       4       7.02
   100       2       3.51
   110       2       3.51
   120       0       0.
   130       0       0.
   140       2       3.51
   150       0       0.
   160       0       0.
   170       0       0.
   180       0       0.
   190       0       0.
   200       0       0.
```

```
SCHEDULING INPUTS
   TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
   16 SVs
    5 GAs
    3 Max contacts
   Actual GA J100 maint, w/ j100 CPU outage
   Pike full availability
   V13 - TBS rank: hP/lI/hST


VALIDATION RESULTS:
                      Number    Max       Min        Mean      Std Dev


START OFFSET:           57      148        0       41.1404   40.9769
GA MATCH:               25


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      20      35.09
    10       7      12.28
    20       3       5.26
    30       2       3.51
    40       4       7.02
    50       2       3.51
    60       6      10.53
    70       0       0.
    80       3       5.26
    90       4       7.02
   100       2       3.51
   110       2       3.51
   120       0       0.
   130       0       0.
   140       2       3.51
   150       0       0.
   160       C       0.
   170       0       0.
   180       0       0.
   190       0       0.
   200       0       0.
```

```
SCHEDULING INPUTS
  TIME: 100/ 0: 0 to 101/ 0: 0 (144000 to 145440)
  16 SVs
   5 GAs
   3 Max contacts
  Actual GA J100 maint, w/ j100 CPU outage
  Pike full availability
  V14 - TBS rank: hP/1I/1ST


VALIDATION RESULTS:
                   Number    Max       Min        Mean      Std Dev


START OFFSET:        57      168        0        41.4211    42.6972
GA MATCH:            25


ACTIVITY START OFFSET:
   Value  Number  Percent

     0      18     31.58
    10       9     15.79
    20       5      8.77
    30       3      5.26
    40       3      5.26
    50       1      1.75
    60       3      5.26
    70       2      3.51
    80       2      3.51
    90       4      7.02
   100       2      3.51
   110       3      5.26
   120       0      0.
   130       0      0.
   140       1      1.75
   150       0      0.
   160       1      1.75
   170       0      0.
   180       0      0.
   190       0      0.
   200       0      0.
```

Appendix D. *Visibility Table Generation.*

This appendix describes the process used to generate the visibility table and prepare it for use with the MCS simulation. The visibility tables used in the MCS simulation were generated using an external program - Pass Scheduler. As the program author, Lt Col T. S. Kelso, describes the program in the instruction manual:

Pass Scheduler is a program which will allow the you to automatically generate schedules of satellite passes from a set of pre-selected files of observation sites and NORAD two-line orbital element sets (6). Pass Scheduler (henceforth PS) requires three type of data to generate visibilities: 1) a description of the satellite orbit; 2) a description of the observer's position; and 3) the viewing parameters. Here is the description of these data and how they were gathered.

D.0.1 Satellite Orbital Data PS requires satellite data be in the form of a North American Air Defense Command (NORAD) two-line orbital element set (commonly abbreviated TLE). NORAD uses this format to distribute satellite orbital data on the space objects tracked by NORAD's Space Surveillance Center. Among other details, a TLE describes the satellite orbit in terms of its Keplerian elements. Below is a description of the NORAD TLE format, from documentation provided with the PS manual.

```
Data for each satellite consists of three lines in the following format:

AAAAAAAAAAA
1 NNNNNU NNNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNNN

Line 0 is a eleven-character name.

Lines 1 and 2 are the standard Two-Line Orbital Element Set Format
identical to that used by NORAD and NASA.  The format description is:

Line 1
```

Column	Description
01-01	Line Number of Element Data
03-07	Satellite Number
10-11	International Designator (Last two digits of launch year)
12-14	International Designator (Launch number of the year)
15-17	International Designator (Piece of launch)
19-20	Epoch Year (Last two digits of year)
21-32	Epoch (Julian Day and fractional portion of the day)
34-43	First Time Derivative of the Mean Motion or Ballistic Coefficient (Depending on ephemeris type)
45-52	Second Time Derivative of Mean Motion (decimal point assumed; blank if N/A)
54-61	BSTAR drag term if GP4 general perturbation theory was used. Otherwise, radiation pressure coefficient. (Decimal point assumed)
63-63	Ephemeris type
65-68	Element number
69-69	Check Sum (Modulo 10) (Letters, blanks, periods, plus signs = 0; minus signs = 1)

Line 2

Column	Description
01-01	Line Number of Element Data
03-07	Satellite Number
09-16	Inclination [Degrees]
18-25	Right Ascension of the Ascending Node [Degrees]
27-33	Eccentricity (decimal point assumed)
35-42	Argument of Perigee [Degrees]
44-51	Mean Anomaly [Degrees]
53-63	Mean Motion [Revs per day]
64-68	Revolution number at epoch [Revs]
69-69	Check Sum (Modulo 10)

All other columns are blank or fixed.

(6)

The TLE describes the orbit at a single instant (the epoch of the TLE), but the position of the satellite at other times can be determined by mathematically "propagating" the epoch trajectory forward or back in time.

TLE's exist for all the GPS satellites now operational. However, the simulation required TLEs for GPS satellites not yet launched. The problem was solved using the trajectories of the existing satellites and the fact that the GPS SVs are spaced at 90 degree intervals in the six orbital planes (5:88). The trajectories for the missing satellites were estimated by propagating the trajectories of the existing satellites to a common epoch. With the satellite's positions fixed at a common time, the missing orbital "slots" in each plane becomes apparent. Assuming the newly launched satellites will fill these holes, element sets were created using the appropriate elements from the existing satellites in the plane, with the locations of the slots filling in the unknown quantities.

Once the GPS SV TLEs have been either gathered or synthesized, they are placed in a file named GPS.TLE to be called by PS at the appropriate time. The TLE file used to generate the visibility is as follows:

```
GPS-0010
1 15271U 84 97  A 92226.07205774 -.00000014  00000-0  99999-4 0  3198
2 15271  62.7700 300.3243 0124103 339.4139  20.1675  2.00582275 56953
GPS BII-12
1 21890U 92  9  A 92226.18328797 -.00000014  00000-0  99999-4 0  1050
2 21890  54.5839 284.3872 0060248 144.1406 216.3366  2.00564032  3484
GPS BII-04
1 20302U 89 85  A 92226.85446246 -.00000015  00000-0  99999-4 0  3997
2 20302  54.0381 284.6837 0016025 327.6882  32.2974  2.00566403 20687
GPS BII-07
1 20533U 90 25  A 92226.67959571 -.00000033  00000-0  99999-4 0  4013
2 20533  55.2252 344.1496 0039438  92.2444 268.1777  2.00563763 17434
GPS BII-02
1 20061U 89 44  A 92226.48398376 -.00000033  00000-0  99999-4 0  4556
2 20061  54.8872 343.8391 0109792 194.7542 164.9389  2.00561894 23309
GPS-0009
1 15039U 84 59  A 92226.62982646 -.00000004  00000-0  99999-4 0  6315
2 15039  63.5674  62.1557 0040952 218.1410 141.6396  2.00568804 59854
GPS BII-13
1 21930U 92 19  A 92225.89620157 -.00000019  00000-0  99999-4 0  1053
2 21930  55.2046  44.8452 0075933 157.9538 202.4065  2.00567847  2405
GPS-0008
1 14189U 83 72  A 92226.46325186 -.00000004  00000-0  99999-4 0  2580
2 14189  63.6084  63.1610 0136525 231.4366 127.4011  2.00563083 66575
GPS-0011
1 16129U 85 93  A 92226.79334660 -.00000003  00000-0  99999-4 0   207
2 16129  64.3249  63.7938 0128101 143.6658 217.2856  2.00565079 50179
GPS BII-11
1 21552U 91 47  A 92226.22305375  .00000006  00000-0  99999-4 0  2105
2 21552  55.4011 104.5586 0042939 225.0711 134.6604  2.00573168  8113
GPS BII-09
1 20830U 90 88  A 92226.54954593  .00000007  00000-0  99999-4 0  3525
2 20830  55.1038 107.1760 0070046 109.7801 251.0560  2.00562367 13938
GPS BII-05
1 20361U 89 97  A 92226.36914852  .00000007  00000-0  99999-4 0  3523
2 20361  55.1884 109.0680 0066568  83.3586 277.4965  2.00563324 10155
GPS BII-01
1 19802U 89 13  A 92226.14753199  .00000016  00000-0  99999-4 0  4509
2 19802  55.0467 166.3474 0041381 172.7861 187.2444  2.00560622 25559
GPS BII 08
1 20724U 90 68  A 92225.99061477  .00000016  00000-0  99999-4 0  3045
2 20724  54.6819 164.8810 0092524 131.8562 228.6309  2.00565479 14842
GPS BII-03
1 20185U 89 64  A 92226.32198841  .00000016  00000-0  99999-4 0  4014
2 20185  54.8802 167.0704 0012047 198.6931 161.2345  2.00557084 21872
GPS BII-10
1 20959U 90103  A 92225.45418467  .00000016  00000-0  99999-4 0  2536
```

```
2 20959  54.9117 166.6623 0062140 219.0715 140.4481  2.00566067 12504
GPS BII-14
1 22014U 92 39   A 92225.93117523  .00000010  00000-0  99999-4 0   231
2 22014  55.0433 224.5891 0083797 275.3527  83.6992  2.00570262   738
GPS BII-06
1 20452U 90  8   A 92226.77349685  .00000009  00000-0  99999-4 0  4005
2 20452  54.1853 223.8693 0044556  65.8465 294.6162  2.00562850 18661
```

D.0.2 Observer Position Data. The positions of the GPS ground stations, especially the Monitor Stations, have been determined very accurately. The positions provided to PS in the file GPS.OBS were taken directly from a GPS Mission Console display. The OBS file is reproduced below.

```
ASCN                1 07 57 04 S 014 24 43 W  105
CAPE                1 28 29 02 N 080 34 22 W  -18
DIEG                1 07 16 12 S 072 22 15 E  -63
KWAJ                1 08 43 23 N 167 43 52 E   41
PIKE                1 38 48 11 N 104 31 30 W 1913
```

Each line describes the location of one ground station. In the Pass Scheduler instructions, Kelso describes the format of the file thus:

> Each observer file must have an extension .OBS. A sample observer file is shown below. Each line allows 20 characters for a place name, a time zone difference (not used), and the observation site's latitude (in degrees, minutes, and seconds followed by N or S), longitude (in degrees, minutes, and seconds followed by W or E), and altitude above mean sea level (in meters). (6)

D.0.3 Pass Schedule Parameters. The remaining information required by PS is to the time difference between the program users' time and Universal Coordinated Time (commonly referred to as "Greenwich" or "ZULU" time), and a number of parameters that define the users' input and output desires. This information is stored in a file called "PASSCHED.CFG", which is shown below.

```
-4                  % Time difference from UTC
N                   % Echo satellite data
```

```
Y                   % Echo schedule to disk file
I:                  % Default drive for support files
                    % Default directory for support files
GPS.OBS             % Default observer database
NEWGPS.TLE          % Default satellite database
```

During program execution, the user is prompted to enter the desired visibility time interval, the satellite elevation (from the ground station local horizon) at which the satellite is declared risen or set, and the minimum duration visibility period the user wishes to have reported.

D.0.4 Pass Schedule Output. In the raw Pass Scheduler output, the name and location of the ground stations for which the following satellite observations apply is listed first, along with the user-defined parameters. Then each satellite is listed in the order in which they rise at that station. The times (under "Rise", "ToC", and "Set") are in hour/minute/second format. "Az" and "El" are satellite azimuth and elevation, respectively. ToC is time of culmination, the instant of maximum elevation for that satellite visibility period. The last column contains codes for classifying the visibility period.

To convert this format to the one needed for the MCS Simulation program, the raw PS output was imported to a computer spreadsheet, the data formed into columns, and the extraneous data removed. The long ground station header was replaced with just the name and index number of the GA.

Bibliography

1. 2 Space Operations Squadron, Falcon Air Force Base, CO. *Navstar Satellite Support Requirements*, 1991.

2. Baker, Kenneth R. *Introduction to Sequencing and Scheduling*. New York: John Wiley & Sons, Inc., 1974.

3. Bellman, Richard, et al. *Mathematical Aspects of Scheduling and Applications*. Oxford: Pergammon Press, 1982.

4. Dale, Lt Keith, GPS Flight Commander. Personal Interviews. 2 Space Operations Squadon-Falcon AFB CO, July 1992.

5. Emshoff, James R. and Roger L. Sisson. *Design and Use of Computer Simulation Models*. New York: Macmillan Publishing Co., Inc., 1970.

6. Headquarters Air Force Space Command, Peterson Air Force Base, CO. *Navstar Global Positioning System (GPS) Systems Operational Requirements Document (SORD)*, January 1990.

7. Kelso, T. S., Lt Col. US Air Force. *Pass Scheduler Program Manual* (1.06 Edition). Assistant Professor, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, July 1990.

8. Kiviat, P. J., et al. *SIMSCRIPT II.5 Programming Language*. CACI Products Company, La Jolla, CA, 1983.

9. Mukherjee, Arup K. "A Simulation Model for Management of Operations in a Pharmacy of a Hospital," *Simulation*, *56*(2):91–103 (Feb 1991).

10. Murrell, Terri. *Introduction to AFIT Educational Computer Services* (5.3 Edition). Air Force Institute of Technology/SCO, Wright-Patterson AFB, OH, May 1991.

11. Navstar-GPS Joint Program Office. *Navstar GPS User Equipment - Introduction*. Technical Report, Los Angeles Air Force Base, CA: US Air Force Space Systems Division, February 1991.

12. Pritsker, A. Alan B. *Introduction to Simulation and SLAM II*. West Lafayette, IN: Systems Publishing Corporation, 1986.

13. Rubin, Kenneth S., et al. "OFMspert: Interface of Operator Intentions in Supervisory Control Using a Blackboard Architecture.," *IEEE Transactions on Systems, Man, and Cybernetics*, *18*:618–637 (July 1988).

14. Serpa, Capt James N., Former GPS Flight Commander. Personal Interview. AFIT/EN Wright-Patterson AFB OH., November 1992.

15. Shannon, Robert E. *Systems Simulation: The Art and Science*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

16. Simon, Herbert A. "Prediction and Prescription is Systems Modeling," *Operations Research*, *38*(1):7–14 (Jan/Feb 1990).

17. Witt, Allison and Drew Goddin. "Health Care Case Study: Simulating Staffing Needs and Work Flow in an Outpatient Diagnostic Center," *Industrial Engineer*, *31*(5):22–26 (May 1989).

Vita

David Nicholas Koster [redacted] attended public school in Toledo, graduating from Jesup W. Scott High School in 1971. After a brief stay at Case Western Reserve University in Cleveland, Ohio, he enlisted in the United States Air Force on 10 November 1972. As an electronic equipment repairman, he served in Mississippi, North Dakota, California, Germany, Mississippi again, and Washington.

In 1980, he returned to Mississippi once again to teach mathematics and basic electronics. From there, as a Technical Sergeant, he was selected for the Airman Education and Commissioning Program. He obtained a BS in Electrical Engineering (Summa Cum Laude) from Wright State University in Dayton, OH in 1984. In that same year he completed Officer Training School as a Distinguished Graduate and was commissioned as a Second Lieutenant in the US Air Force. He then served three years as a Communications Plans and Architectures officer at Langley AFB, Virginia.

Selected to enter the Space Operations career field, he completed Undergraduate Space Training in 1988 as a Distinguished Graduate. Just prior to attending the Air Force Institute of Technology, he served as the Chief, Navigation Analysis Section for the 2nd Space Operations Squadron, the operators of the Navstar Global Positioning System.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

[illegible]

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE December 1992

3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE

A MODEL OF GLOBAL POSITIONING SYSTEM (GPS) MASTER CONTROL SYSTEM (MCS) OPERATIONS

5. FUNDING NUMBERS

6. AUTHOR(S)

David N. Koster, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSO/ENS/92D-9

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Abstract

The United States Air Force's Navstar Global Positioning System (GPS) provides high-accuracy space-based navigation and time distribution to suitably-equipped military and civilian users. The system consists of earth-orbiting satellites and a world-wide network of ground stations. A single operational control center, the GPS Master Control Station (MCS) monitors, maintains, and commands the GPS satellite constellation. The on-going deployment of the complete satellite constellation and recent changes in the operational crew structure may invalidate previously used planning and management paradigms. There is currently no analytical method for predicting the impact of these and other environmental changes on system parameters and performance. Extensive testing cannot be performed at the MCS itself due to the criticality of the GPS mission and lack of operational redundancy. This research provides and validates a discret event simulation model of the MCS operations center task flow, focusing on the creation and testing of a sliding-widow MCS activity scheduler. The simulation was validated using MCS historical data. Experiments were conducted by varying the number of ground stations and satellite constellation size available to the simulation. The results, while not quantitatively trustworthy, were used to draw general conclusions about the GPS operational environment.

14. SUBJECT TERMS

Navstar, Global Positioning System, GPS, operations control center, Master Control Station, MCS, activity scheduling, simulation.

15. NUMBER OF PAGES

207

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL